

SKRIPSI

KLASIFIKASI BAKTERI DENGAN METODE DEEP LEARNING

Disusun dan diajukan oleh

ALIF TRI HANDOYO

D42116504



DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

LEMBAR PENGESAHAN SKRIPSI

KLASIFIKASI BAKTERI DENGAN METODE DEEP LEARNING

Disusun dan diajukan oleh:

ALIF TRI HANDOYO

D421 16 504

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin
pada tanggal 1 Februari 2021
dan dinyatakan telah memenuhi syarat kelulusan

menyetujui,

Pembimbing Utama,



Dr. Ir. Ingrid Nurtanio, M.T.
NIP. 19610813 198811 2 001

Pembimbing Pendamping,



Anugravani Bustamin, S.T., M.T.
NIP. 19901201 201807 4 001

Ketua Program Studi,



Dr. Amil Ahmad Ilham, S.T., M.IT
NIP. 19731010 199802 1 001

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Alif Tri Handoyo
NIM : D421 16 504
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul :

KLASIFIKASI BAKTERI DENGAN METODE DEEP LEARNING

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 1 Februari 2021

Yang menyatakan,



(Alif Tri Handoyo)

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya sehingga skripsi yang berjudul “**Klasifikasi Bakteri dengan Metode *Deep Learning***” ini dapat terselesaikan dengan baik sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa banyak kendala yang dihadapi dalam menyelesaikan skripsi ini, namun berkat dorongan, dukungan, bimbingan, serta motivasi yang diberikan sehingga penyusunan skripsi ini dapat terselesaikan dengan baik. Ucapan terima kasih serta penghargaan yang setinggi-tingginya penulis sampaikan kepada:

1. Tuhan Yang Maha Esa yang melalui berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan tugas akhir ini.
2. Kedua Orang tua penulis, Bapak Ir. Hadi Purwanto, M.T., dan Ibu St. Syarfiah yang selalu memberikan dukungan, doa, semangat dan kasih sayang serta selalu sabar dalam mendidik penulis sejak kecil.
3. Ibu Dr. Ir. Ingrid Nurtanio, M.T., selaku pembimbing I dan Ibu Anugrayani Bustamin, S.T., M.T., selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran dan memberikan bimbingan dalam penyusunan skripsi ini.
4. Bapak Dr. Amil Ahmad Ilham, S.T., M.IT., selaku ketua Prodi Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas ilmu, bimbingan dan arahnya selama masa perkuliahan penulis.

5. Segenap Dosen dan Staff Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah banyak membantu penulis selama masa perkuliahan.
6. Teman-teman Laboratorium *AIMP Research Group* FT-UH yang telah memberikan bantuan selama penelitian, dan diskusi mengenai penyusunan skripsi.
7. Teman-teman Prodi Teknik Informatika angkatan 2016 atas dukungan dan semangat yang telah diberikan.
8. Serta berbagai pihak atas segala dukungan dan bantuannya yang tidak dapat penulis tuliskan satu persatu.

Akhir kata, penulis berharap semoga Allah SWT. membalas segala kebaikan dari semua pihak yang telah membantu penulis dalam penyusunan skripsi ini, dan semoga skripsi ini dapat memberikan hal yang bermanfaat serta menambah wawasan ilmu untuk pembaca dan juga bagi penulis sendiri.

Makassar, 1 Februari 2021

A handwritten signature in black ink, featuring a stylized star-like symbol above the name 'Alif' and a horizontal line underneath.

Penulis,

(Alif Tri Handoyo)

ABSTRAK

Mikroorganisme seperti bakteri menjadi penyebab utama berbagai penyakit infeksi seperti kolera, botulisme, gonore, penyakit lyme, radang tenggorokan, TBC dan sebagainya. Oleh karena itu, identifikasi dan klasifikasi bakteri sangat penting dalam dunia kedokteran untuk membantu para ahli mendiagnosa penyakit yang diderita oleh pasien. Namun, untuk identifikasi dan klasifikasi bakteri secara manual dibutuhkan individu profesional dan waktu yang lama. Dengan bantuan kecerdasan buatan kita dapat secara efektif dan efisien mengklasifikasikan bakteri dan menghemat banyak waktu serta tenaga manusia. Dalam penelitian ini, dibuat sistem untuk mengklasifikasikan bakteri dari sampel gambar mikroskopis. Sistem ini menggunakan *deep learning* dengan metode *transfer learning*. Arsitektur Inception V3 dimodifikasi dan *training* ulang menggunakan 108 sampel citra dengan *label* lima jenis bakteri yaitu *Acinetobacter baumannii*, *Escherichia coli*, *Neisseria gonorrhoeae*, *Propionibacterium acnes* dan *Veionella*. Data kemudian dibagi menjadi *training* dan *validation* dengan metode *k-fold cross validation*. Selanjutnya, fitur yang telah diekstraksi oleh model di *training* dengan konfigurasi *minibatchsize 5*, *maxepoch 5*, *initiallearnrate 0,0001*, dan *validationfrequency 3*. Model kemudian diuji dengan data *validation* dengan melakukan sepuluh kali percobaan dan mendapatkan nilai rata-rata akurasi sebesar 94.43% dan rata-rata waktu eksekusi selama 44 detik.

Kata kunci: Klasifikasi bakteri, *Deep Learning*, *Inception V3*, *Transfer Learning*.

ABSTRACT

Microorganisms such as bacteria are the main cause of various infectious diseases such as cholera, botulism, gonorrhea, Lyme disease, sore throat, tuberculosis and so on. Therefore, identification and classification of bacteria is very important in the world of medicine to help experts diagnose diseases suffered by patients. However, manual identification and classification of bacteria takes a long time and a professional individual. With the help of artificial intelligence we can effectively and efficiently classify bacteria and save a lot of time and human labor. In this study, a system was created to classify bacteria from microscopic image samples. This system uses deep learning with the transfer learning method. Inception V3 architecture was modified and retained using 108 image samples labeled with five types of bacteria, namely *Acinetobacter baumannii*, *Escherichia coli*, *Neisseria gonorrhoeae*, *Propionibacterium acnes* and *Veionella*. The data is then divided into training and validation using the k-fold cross validation method. Furthermore, the features that have been extracted by the model are trained with the configuration of minibatchsize 5, maxepoch 5, initiallearnrate 0.0001, and validation frequency 3. The model is then tested with data validation by conducting ten experiments and getting an average accuracy value of 94.43% and average execution time of 44 seconds.

Keywords: Bacterial Classification, Deep Learning, Inception V3, Transfer Learning.

DAFTAR ISI

KATA PENGANTAR	i
ABSTRAK	iii
DAFTAR ISI	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	3
1.4. Manfaat Penelitian	3
1.5. Batasan Masalah Penelitian.....	3
1.6. Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1. Pengertian Bakteri	6
2.2. Sejarah Klasifikasi Bakteri.....	6
2.3. Metode Klasifikasi Bakteri	8
2.3.1. Klasifikasi Berdasarkan Bentuk	9
2.3.2. Klasifikasi Berdasarkan Pewarnaan Gram.....	10
2.3.3. Klasifikasi Berdasarkan Suhu	12

2.3.4.	Klasifikasi Berdasarkan Kebutuhan Oksigen.....	13
2.4.	<i>Artificial Neural Network</i>	14
2.5.	<i>Deep Neural Network</i>	16
2.6.	<i>Convolutional Neural Network</i>	17
2.6.1.	<i>Convolution Layer</i>	18
2.6.2.	<i>Weights dan Bias</i>	20
2.6.3.	Fungsi Aktivasi	21
2.6.4.	<i>Pooling Layer</i>	24
2.6.5.	<i>Fully Connected Layer</i>	25
2.7.	<i>Inception V3</i>	25
1.	<i>Factorizing Convolutions</i>	26
2.	<i>Auxiliary Classifier</i>	30
3.	<i>Efficient Grid Size Reduction</i>	30
4.	Arsitektur <i>InceptionV3</i>	31
2.8.	<i>K-Fold Cross Validation</i>	33
2.9.	<i>Grad-CAM</i>	34
2.10.	<i>Data Augmentation</i>	36
BAB III METODOLOGI PENELITIAN.....		38
3.1.	Tahapan Penelitian	38
3.2.	Waktu dan Lokasi Penelitian	39
3.3.	Instrumen Penelitian.....	39

3.4.	Teknik Pengambilan Data	39
3.5.	Perancangan Sistem	40
3.5.1.	Proses <i>Training</i>	41
3.5.2.	Proses Validation	57
3.5.3.	Proses Testing	61
3.6.	Perbandingan Sistem InceptionV3 dengan Tujuh Arsitektur Lain	64
3.7.	Analisis Kerja Sistem	65
BAB IV HASIL DAN PEMBAHASAN		66
4.1.	Hasil Penelitian	66
4.1.1.	Hasil Percobaan Pertama.....	67
4.1.2.	Hasil Percobaan Kedua	71
4.1.3.	Hasil Percobaan Ketiga	74
4.1.4.	Hasil Percobaan Keempat	76
4.1.5.	Percobaan Menggunakan Data <i>Training</i> Sebagai Data Validasi	78
4.2.	Pembahasan.....	79
BAB V PENUTUP.....		89
5.1.	Kesimpulan	89
5.2.	Saran.....	89
DAFTAR PUSTAKA		90
LAMPIRAN.....		92

DAFTAR GAMBAR

Gambar 2.1 Klasifikasi Bakteri Berdasarkan Bentuk	10
Gambar 2.2 Klasifikasi Bakteri Berdasarkan Pewarnaan Gram	12
Gambar 2.3 Klasifikasi Bakteri Berdasarkan Suhu.....	13
Gambar 2.4 Klasifikasi Bakteri Berdasarkan Kebutuhan Oksigen	14
Gambar 2.5 Visualisasi <i>Artificial Neural Network</i>	15
Gambar 2.6 Visualisasi <i>Deep Neural Network</i>	17
Gambar 2.7 Ilustrasi Arsitektur <i>Convolutional Neural Network</i> (CNN)	18
Gambar 2.8 Ilustrasi Proses <i>Convolutional Layer</i>	19
Gambar 2.9 Ilustrasi <i>Weights</i> dan <i>Bias</i> pada <i>Layer</i>	20
Gambar 2.10 Grafik Fungsi Aktivasi ReLU	23
Gambar 2.11 Ilustrasi fungsi aktivasi <i>softmax</i>	24
Gambar 2.12 Operasi <i>Max Pooling</i>	25
Gambar 2.13 Dua konvolusi 3x3 digantikan oleh konvolusi 5x5	27
Gambar 2.14 <i>Inception Module A</i> dengan <i>factorization</i>	27
Gambar 2.15 Satu konvolusi 3x1 yang diikuti konvolusi 1x3 diganti oleh konvolusi 3x3	28
Gambar 2.16 <i>Inception Module B</i> dengan <i>asymmetric factorization</i>	29
Gambar 2.17 <i>Inception Module C</i> dengan <i>asymmetric factorization</i>	29
Gambar 2.18 <i>Auxiliary Classifier</i> sebagai <i>regularization</i>	30
Gambar 2.19 <i>Conventional downsizing</i>	31
Gambar 2.20 Arsitektur <i>InceptionV3</i>	32
Gambar 2.21 Contoh <i>5 fold cross validation</i>	33
Gambar 2.22 Visualisasi <i>Grad-CAM</i>	34

Gambar 3.1 Diagram Tahapan Penelitian	38
Gambar 3.2 <i>Flowchart</i> Perancangan Sistem	40
Gambar 3.3 Contoh Data Latih	41
Gambar 3.4 Hasil Modifikasi <i>Layer InceptionV3</i>	43
Gambar 3.5 Jumlah <i>dataset</i> setiap kelas	45
Gambar 3.6 Pembagian data <i>training</i> dan <i>validation</i> menggunakan <i>k-fold cross validation</i>	45
Gambar 3.7 <i>Flowchart</i> Proses <i>K-fold Cross Validation</i>	46
Gambar 3.8 Hasil <i>augmentasi</i> dengan parameter berbeda.....	49
Gambar 3.9 Konfigurasi <i>Augmentasi</i> pada Setiap <i>Data Training</i>	49
Gambar 3.10 Visualisasi Proses Konvolusi dan <i>Feature mapping</i>	51
Gambar 3.11 <i>Graph</i> Proses <i>Training</i>	55
Gambar 3.12 Visualisasi <i>Grad-CAM</i>	60
Gambar 3.13 GUI (<i>Graphical User Interface</i>)	62
Gambar 3.14 Visualisasi Data Testing.....	63
Gambar 3.15 <i>Flowchart</i> Program Perbandingan InceptionV3 dengan tujuh arsitektur lain.....	64
Gambar 4.1 <i>Confusion</i> Matriks Arsitektur <i>InceptionV3</i> dengan <i>K-fold cross validation</i>	68
Gambar 4.2 Nilai Awal <i>Weight</i> dan <i>Bias</i> Model Terbaik	70
Gambar 4.3 Klasifikasi Data Testing Percobaan Pertama	71
Gambar 4.4 <i>Confusion</i> Matriks Arsitektur <i>InceptionV3</i> tanpa <i>K-fold cross validation</i>	72
Gambar 4.5 Klasifikasi Data Testing Percobaan Kedua	73

Gambar 4.6 <i>Confusion</i> Matriks Arsitektur <i>InceptionV3</i> dengan konfigurasi <i>RNG default</i>	74
Gambar 4.7 Klasifikasi Data Testing Percobaan Ketiga.....	75
Gambar 4.8 <i>Confusion</i> Matriks Arsitektur <i>Inceptionv3</i> dengan <i>k-fold cross validation</i> tanpa <i>augmentation</i>	76
Gambar 4.9 Klasifikasi Data Testing Percobaan Keempat.....	78
Gambar 4.10 <i>Confusion Matriks</i> Percobaan Data <i>Training</i> Sebagai Data Validasi	79

DAFTAR TABEL

Tabel 2.1 Sejarah Klasifikasi Bakteri.....	6
Tabel 2.2 Parameter Arsitektur <i>InceptionV3</i>	32
Tabel 3.1 Parameter Konvolusi <i>Transfer Learning-based model InceptionV3</i>	52
Tabel 4.1 Hasil Prediksi Sistem <i>InceptionV3</i> dengan <i>k-fold cross validation</i>	67
Tabel 4.2 Validasi <i>InceptionV3</i> dengan <i>K-fold cross validation</i>	69
Tabel 4.3 Validasi <i>InceptionV3</i> tanpa <i>K-fold cross validation</i>	72
Tabel 4.4 Validasi <i>InceptionV3</i> dengan konfigurasi <i>RNG Default</i>	74
Tabel 4.5 Validasi <i>InceptionV3</i> tanpa <i>augmentation</i>	77
Tabel 4.6 Hasil Kesalahan Klasifikasi	81
Tabel 4.7 Hasil Klasifikasi Benar	81
Tabel 4.8 Perbandingan <i>Inceptionv3</i> dengan Tujuh Arsitektur Lain.....	88

BAB I

PENDAHULUAN

1.1. Latar Belakang

Menurut *Food and Agriculture Organization* (FAO), setiap tahunnya jumlah korban meninggal akibat infeksi bakteri mencapai hingga 700.000 jiwa. Pengenalan genera dan spesies bakteri dibutuhkan karena pengetahuan biologis mikroorganisme sangat penting dalam ilmu kedokteran, kedokteran hewan, biokimia, industri makanan atau pertanian. Meskipun sebagian besar mikroorganisme berdampak positif pada berbagai bidang kehidupan, mereka dapat menjadi penyebab banyak penyakit, termasuk penyakit menular. (Zieliński *et al.*, 2017).

Ahli biologi mengidentifikasi dan mengklasifikasikan berbagai jenis bakteri yang memiliki biokimia dan bentuk yang berbeda. Mereka menggunakan atribut bakteri yang berbeda untuk klasifikasi. Misalnya, bentuk sel bakteri (spiral, silindris, dan bola), ukuran dan struktur koloni yang dibentuk oleh bakteri diperiksa untuk membedakan spesies bakteri. Sel dari beberapa jenis bakteri memiliki ukuran dan struktur yang berbeda tergantung pada kondisi lingkungan. Beberapa spesies bakteri memiliki bentuk yang sangat mirip. Meskipun setiap spesies bakteri memiliki karakteristiknya sendiri, reaksi biokimia yang dilakukan oleh bakteri dan aktivitas metabolisme yang mereka lakukan bersama-sama membantu untuk mengklasifikasikan spesies. Namun, klasifikasi spesies bakteri bukanlah tugas yang mudah bahkan bagi seorang spesialis yang berpengalaman (Talo, 2019).

Secara umum, analisis citra mikrobiologi dengan metode laboratorium tradisional terdapat kesalahan pengenalan bakteri, dan membutuhkan pengalaman ekstra serta waktu pengerjaan yang lama. Oleh karena itu, teknik klasifikasi otomatis citra bakteri lebih bermanfaat dibanding pengamatan visual tradisional bagi ahli biologi karena pengklasifikasian yang akurat, biaya rendah, dan diagnosis cepat (Mohamed dan Afify, 2019).

Penelitian sebelumnya terkait klasifikasi bakteri dilakukan pada tahun 2018 oleh basma, dkk melakukan penelitian dengan mengklasifikasi sepuluh macam bakteri menggunakan *feature extraction Bag of words* serta metode *Support Vector Machine* (SVM) dengan hasil akurasi 97%.

Pada tahun 2019 Treesukon, dkk melakukan penelitian dengan mengklasifikasi dua macam bakteri dengan metode *deep learning* berbasis *python* dan menggunakan arsitektur LeNET dengan hasil akurasi lebih dari 75 persen. Pada tahun 2018 lei huang menggunakan metode CNN dengan arsitektur AlexNET untuk mengklasifikasi 18 kategori bakteri dengan hasil akurasi 73%.

Pada penelitian ini penulis akan membuat sistem yang mengklasifikasi bakteri dengan metode *deep learning* CNN menggunakan arsitektur *InceptionV3*. Arsitektur *InceptionV3* digunakan karena dalam penelitian sebelumnya banyak digunakan untuk mengklasifikasi dengan hasil akurasi yang cukup tinggi seperti klasifikasi penderita penyakit covid-19 (Asif *et al.*, 2020), dan juga klasifikasi penderita penyakit paru-paru (Wang *et al.*, 2019).

1.2. Rumusan Masalah

Rumusan masalah yang akan diuraikan dalam skripsi ini antara lain:

1. Bagaimana proses klasifikasi bakteri dari sebuah *dataset image* menggunakan algoritma CNN dengan arsitektur *InceptionV3*?
2. Bagaimana unjuk kerja algoritma CNN dengan arsitektur *InceptionV3* dari sisi akurasi dan waktu proses *training* pada klasifikasi bakteri?

1.3. Tujuan Penelitian

Tujuan akhir dari penelitian ini antara lain:

1. Untuk melihat hasil proses klasifikasi bakteri dari sebuah *dataset image* menggunakan algoritma CNN dengan arsitektur *InceptionV3*.
2. Untuk mengetahui unjuk kerja algoritma CNN dengan arsitektur *InceptionV3* dari sisi akurasi dan waktu proses *training* dalam mengklasifikasikan bakteri.

1.4. Manfaat Penelitian

Dengan dilakukannya penelitian ini, diharapkan manfaat yang didapatkan antara lain:

1. Bagi masyarakat, penelitian ini diharapkan secara tidak langsung dapat membantu masyarakat dengan membantu perkembangan dari sistem kesehatan.
2. Bagi peneliti, penelitian ini dapat dijadikan referensi terkait kemampuan arsitektur *InceptionV3* dalam mengklasifikasi bakteri.
3. Bagi institusi pendidikan, penelitian ini dapat digunakan sebagai referensi ilmiah untuk penelitian-penelitian selanjutnya.

1.5. Batasan Masalah Penelitian

Yang menjadi Batasan masalah dalam skripsi ini adalah:

1. Data yang digunakan diambil dari misztal.edu.pl/software/databases/dibas/.
2. *Dataset* penelitian terdiri dari lima macam bakteri berbahaya menurut *CDC's Antibiotic Resistance Threats in the United States, 2019 (2019 AR Threats Report)*, yaitu bakteri *Acinetobacter baumannii*, *Escherichia coli*, *Neisseria gonorrhoeae*, *Propionibacterium acnes* dan *Veionella*.
3. Sistem Klasifikasi dibuat menggunakan *Matlab R2020b* dengan menggunakan arsitektur *InceptionV3*.

1.6. Sistematika Penulisan

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu :

BAB I PENDAHULUAN

Bab ini menguraikan secara umum mengenai hal yang menyangkut latar belakang, perumusan masalah dan bahasan masalah, tujuan, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori tentang hal-hal yang berhubungan dengan klasifikasi bakteri, visi komputer, pemrosesan citra, *deep learning*, dan metode yang digunakan.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang tahap penelitian, instrumen penelitian, dan penerapan algoritma serta teknik pengolahan data.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dan pengolahan data serta pembahasan yang disertai tabel hasil penelitian.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1. Pengertian Bakteri

Bakteri merupakan penyebab utama berbagai penyakit. Hingga pertengahan abad ke-20, penyakit infeksi bakteri menjadi penyebab utama kematian di kalangan lansia. Sanitasi yang lebih baik, vaksin, dan antibiotik telah berhasil menurunkan tingkat kematian akibat infeksi bakteri, meskipun seiring berjalannya waktu, ada beberapa jenis bakteri yang mulai kebal terhadap antibiotik sehingga menyebabkan penyakit yang ditimbulkannya semakin sulit untuk diobati. Bakteri menyebabkan penyakit dengan mengeluarkan racun (seperti dalam botulisme), dengan memproduksi racun secara internal, yang dilepaskan saat bakteri membelah diri (seperti pada bakteri penyebab penyakit tipus), atau dengan menginduksi kepekaan terhadap sifat antigeniknya (seperti pada bakteri penyebab penyakit tuberkulosis). Penyakit bakteri serius lainnya termasuk kolera, difteri, meningitis bakterial, tetanus, penyakit Lyme, gonore, dan sifilis (Britannica, 2018).

2.2. Sejarah Klasifikasi Bakteri

Tabel 2.1 Sejarah Klasifikasi Bakteri

Rentang waktu (Tahun)	Klasifikasi utama berdasarkan
Akhir abad ke -19	<i>Morphology, Growth Requirements, Pathogenic potential.</i>
1900 – 1960	<i>Morphology, Physiology, Biochemistry.</i>

Rentang waktu (Tahun)	Klasifikasi utama berdasarkan
1960 – 1980	<i>Chemotaxonomy, Numerical Taxonomy, DNA–DNA Hybridization.</i>
1980 – sekarang	<i>Genotypic Analyses, Multilocus Sequence Analyses, Average Nucleotide Identity, Whole Genome Analysis.</i>

Sejarah dari klasifikasi bakteri dengan jelas menunjukkan bahwa perubahan disebabkan oleh muncul nya teknik - teknik baru dalam proses klasifikasi, hal ini dapat dilihat diatas pada Tabel 2.1. Akhir abad ke-19 adalah awal taksonomi bakteri dan Ferdinand Cohn pada tahun 1872 adalah orang pertama yang mengklasifikasikan enam genera bakteri (sebagai anggota tumbuhan) yang sebagian besar berdasarkan morfologi mereka. Namun, saat itu mayoritas ilmuwan tertarik dengan gambaran bakteri patogen. Sebenarnya, banyak bakteri patogen yang dikenal saat ini ditemukan antara tahun 1880 dan 1900. Pada saat itu, selain morfologi, persyaratan pertumbuhan dan potensi patogen merupakan penanda taksonomi yang paling penting.

Pada awal abad ke-20 semakin banyak data fisiologis dan biokimia yang digunakan, selain morfologi sebagai penanda penting untuk klasifikasi dan identifikasi mikroorganisme, berbagai sifat biokimia dan fisiologis kultur bakteri ditentukan untuk karakterisasi dan identifikasi mereka. Kemudian, enzim dipelajari

dan jalur metabolisme dijelaskan. Edisi pertama dari *Bergey's Manual of Determinative Bacteriology* mengklasifikasikan bakteri pada tahun 1923 atas dasar sifat fenotipik sebagai "tipikal tumbuhan *unicellular*", yang disebut *schizomycetes*. Bahkan dalam *bergey's manual* edisi ke-7 yang diterbitkan pada tahun 1957, bakteri masih diklasifikasikan sebagai anggota tumbuhan (*protophyta*, tumbuhan primitif). Berdasarkan sekuens parsial dari gen *16S ribosomal RNA* (rRNA), *Archaea* (awalnya bernama *Archaeobacteria*) pertama kali diklasifikasikan sebagai kategori terpisah pada tahun 1977.

Protistolog Prancis Edouard Chatton, mentor dan teman lama A. Lwoff, menyebutkan untuk pertama kalinya pada tahun 1925 dua kategori *prokariota* dan *eukariota* tetapi hanya untuk membedakan *prokariotik* dari *protista eukariotik* saja. Namun, proposalnya tidak diketahui secara umum. Kemudian, A. Lwoff menyebarkan perbedaan ini dan akhirnya meyakinkan R. Stanier, bersama dengan C.B. van Niel pada tahun 1962, untuk menjelaskan pembagian organisme *prokariotik* (bakteri) dan *eukariotik* (hewan, tumbuhan) yang rinci dan diterima dengan baik. Dalam *bergey's manual* edisi ke-8, yang diterbitkan pada tahun 1974, bakteri tidak lagi dianggap sebagai tumbuhan dan diakui sebagai anggota kategori *Procaryotae*. Namun, semua gagasan sebelumnya tentang *filogeni* dan hubungannya dibuang dan bakteri diatur dalam kelompok berdasarkan pada pewarnaan gram, morfologi dan kebutuhan oksigen (Schleifer, 2009).

2.3. Metode Klasifikasi Bakteri

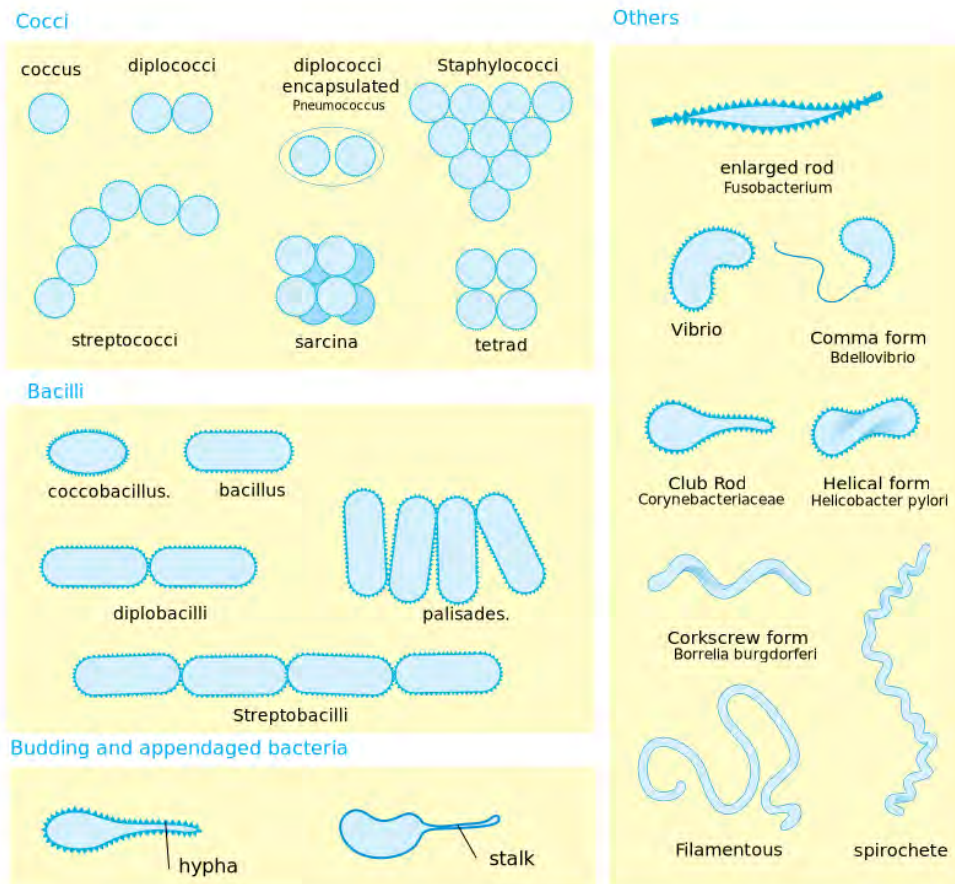
Bakteri dapat diklasifikasikan berdasarkan bentuk, pewarnaan Gram, suhu, dan kebutuhan oksigennya. Berikut penjelasannya:

2.3.1. Klasifikasi Berdasarkan Bentuk

Jika dilihat dari bentuknya, bakteri dapat dibagi menjadi bakteri *coccus*, *bacilli*, dan *spiral*. *Coccus* memiliki bentuk bulat atau bujur telur. Jenis yang satu ini bisa hidup sendiri, tapi bisa juga hidup dalam formasi dengan bakteri sejenis lainnya. Dua *coccus* yang bergabung disebut *diplococci*, sementara empat *coccus* yang membentuk kotak disebut *tetrad*. Susunan yang umum didapatkan dari jenis ini adalah rantai bakteri yang disebut *streptococci*.

Bacilli atau *bacillus* adalah kelompok yang membentuk batang, walaupun nama *bacillus* juga digunakan sebagai *genus*. Kebanyakan bakteri di kategori ini berbentuk batang tunggal, walaupun ada juga *diplobacilli* yang muncul berpasangan dan *streptobacilli* yang muncul berantai.

Terakhir, *spiral* atau *spirochetes* adalah bakteri yang membentuk melengkung. Banyak jenis ini yang sifat tubuhnya kaku dan punya kemampuan untuk bergerak. Jenis spiral dibagi lagi menjadi *vibrio*, *spirilla*, dan *spirochetes*. *Vibrio* berbentuk seperti karakter koma, sementara *spirilla* punya struktur spiral yang kaku. Terakhir, *Spirochetes* membentuk spiral yang fleksibel (Putri, 2017). Ilustrasi bentuk bakteri ini dapat dilihat pada Gambar 2.1.



Gambar 2.1 Klasifikasi Bakteri Berdasarkan Bentuk (Putri, 2017).

2.3.2. Klasifikasi Berdasarkan Pewarnaan Gram

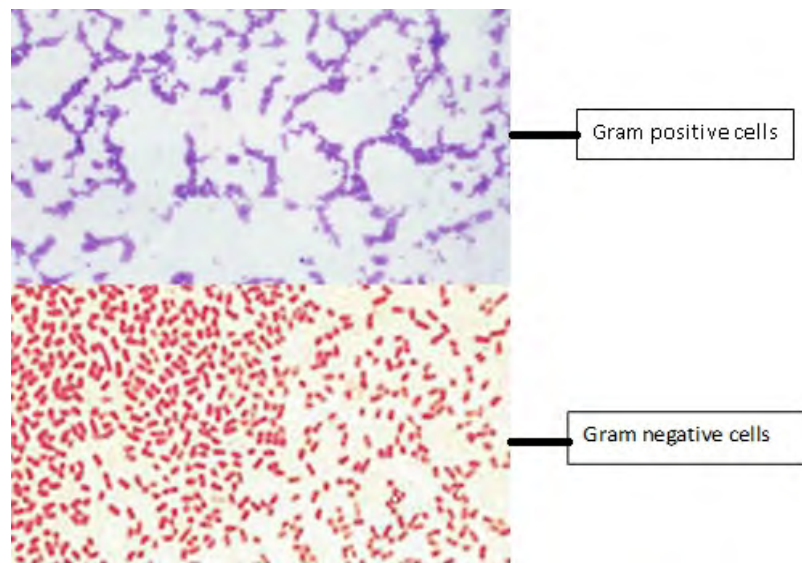
Pewarnaan Gram merupakan pewarnaan yang sangat umum dalam bidang bakteriologi. Dengan pewarnaan ini, kelompok bakteri dapat dibedakan menjadi dua, yaitu kelompok bakteri Gram positif dan bakteri Gram negatif. Ada banyak modifikasi dari teknik pewarnaan ini, namun semua teknik tersebut berdasarkan pada prinsip yang sama, yaitu :

1. Mewarnai mikroorganisme dengan pewarna dasar, yaitu dengan kristal violet atau gentian violet.

2. Fiksasi warna, yaitu untuk menguatkan perlekatan warna dasar, misalnya dilakukan dengan garam iodine (modifikasi larutan lugol).
3. Pencucian atau penghapusan warna dasar dengan alkohol, aseton, atau campuran alkohol dengan aseton.
4. Pewarnaan kembali dengan pewarna pembanding atau kontras yang berbeda dengan pewarna dasar, yaitu untuk mewarnai sel-sel yang telah hilang warnanya oleh penghapusan warna. Misalnya dalam hal ini dipakai safranin atau karbol fuhsin.

Bakteri yang setelah diwarnai dengan pewarna dasar warnanya tidak terhapus oleh alkohol akan berwarna violet karena terwarnai oleh kristal violet, dan tidak lagi menyerap pewarna kontras. Kelompok bakteri yang mempunyai sifat ini dikelompokkan menjadi bakteri gram positif. Sedangkan, kelompok bakteri yang telah diwarnai dengan pewarna dasar dan warnanya terhapus setelah diperlakukan dengan alkohol, akan menyerap pewarna safranin atau karbol fuhsin yang dipakai sebagai pewarna kontras, sehingga dalam preparat akan terlihat warna merah (warna safranin atau karbol fuhsin). Kelompok bakteri yang demikian disebut dengan bakteri gram negatif (Sujaya, 2016).

Ilustrasi bakteri melalui pewarnaan gram dapat dilihat pada Gambar 2.2.



Gambar 2.2 Klasifikasi Bakteri Berdasarkan Pewarnaan Gram (Putri, 2017).

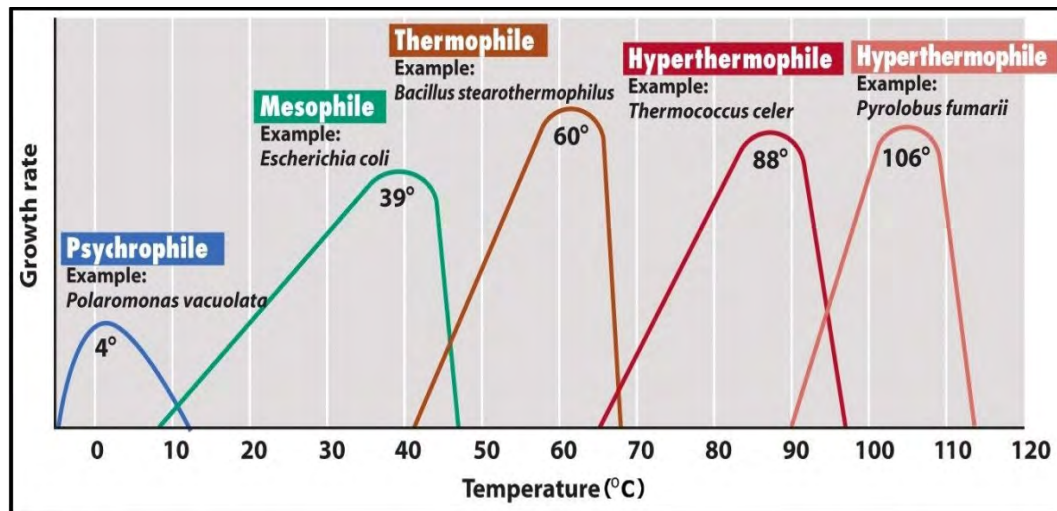
2.3.3. Klasifikasi Berdasarkan Suhu

Berdasarkan kemampuan adaptasinya terhadap suhu, bakteri terbagi menjadi tiga jenis, yaitu *thermophile*, *mesophile*, dan *psychrophile*. *Thermophile* mampu bertahan di lingkungan bersuhu tinggi, yaitu 41-122 derajat celsius. Jenis ini biasanya ditemukan di wilayah hangat di bumi, seperti mata air panas, lautan dalam *hidrotermal*, dan *kompos*.

Mesophile adalah jenis yang pertumbuhan optimalnya berada di suhu sedang, yaitu 20-45 derajat celsius. Biasanya, jenis ini terdapat pada keju dan yogurt. Sebagian besar patogen yang menyerang manusia juga termasuk pada *mesophile*.

Psychrophile adalah kelompok bakteri yang dapat tumbuh dan bereproduksi di suhu dingin, dari 20-10 derajat celsius. Ciri-cirinya adalah membran sel lipid yang secara kimia tahan terhadap suhu dingin dan sering membuat protein antibeku

untuk menjaga cairan *internal* dan DNA mereka (Putri, 2017). Ilustrasi klasifikasi bakteri berdasarkan suhu dapat dilihat pada Gambar 2.3.



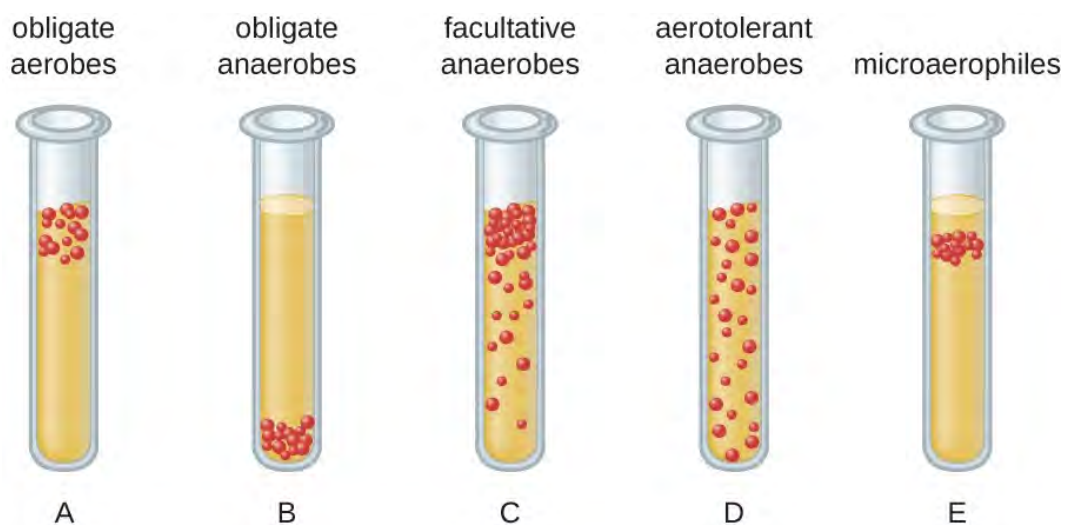
Gambar 2.3 Klasifikasi Bakteri Berdasarkan Suhu (Putri, 2017).

2.3.4. Klasifikasi Berdasarkan Kebutuhan Oksigen

Berdasarkan kebutuhannya akan oksigen, bakteri dibagi menjadi *aerobik*, *anaerobik*, *anaerobik fakultatif*, dan *micro-aerophilic*. *Aerobik* adalah bakteri yang memerlukan oksigen untuk proses metabolisme dan respirasi seluler. Jenis ini menggunakan oksigen untuk melakukan metabolisme senyawa seperti karbohidrat dan lemak untuk menghasilkan energi. Jenis ini juga dapat menghasilkan lebih banyak energi ATP daripada respirasi anaerobik atau fermentasi, tapi ia rawan terkena stres oksidatif.

Anaerobik dibagi menjadi tiga, yaitu *anaerobik obligat*, *anaerobik aerotoleran*, dan *anaerobik fakultatif*. *Anaerobik obligat* adalah jenis yang tidak membutuhkan oksigen untuk pertumbuhannya. Malah, jenis ini bisa mati jika terkena oksigen, walaupun dengan kadar yang berbeda-beda.

Anaerobik fakultatif membuat energi ATP dengan respirasi aerobik jika terdapat oksigen di lingkungannya, tapi bisa berganti menjadi respirasi anaerobik atau fermentasi jika tidak ada oksigen. Terakhir, *micro-aerophilic* adalah jenis yang membutuhkan oksigen untuk bertahan hidup, tapi hanya dalam konsentrasi rendah. Bakteri jenis ini memerlukan oksigen karena tidak mampu melakukan fermentasi ataupun respirasi anaerobik. Tapi, jenis ini akan teracuni jika terpapar konsentrasi oksigen yang tinggi (Putri, 2017). Untuk mengklasifikasi bakteri berdasarkan kebutuhan oksigennya, bakteri di kultur menggunakan tabung tioglikolat, ilustrasi nya dapat dilihat pada Gambar 2.4.



Gambar 2.4 Klasifikasi Bakteri Berdasarkan Kebutuhan Oksigen (Putri, 2017)

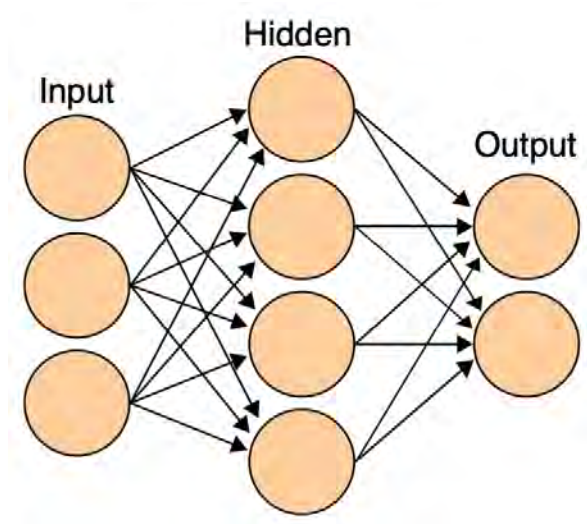
2.4. *Artificial Neural Network*

Artificial Neural Network atau ANN adalah model pembelajaran yang terinspirasi dari sistem biologis pembelajaran dan klasifikasi. Secara sederhana, ANN bekerja pada banyak lapisan yang terhubung melalui *neuron*:

- **Input Layer:** Dimana data diberikan sebagai *input*.

- **Hidden Layer:** Data *input* kemudian melewati neuron berdasarkan fungsi aktivasi.
- **Output Layer:** *Output* dihasilkan di sini.

Secara visual kita dapat menggambarkan ANN seperti pada Gambar 2.5



Gambar 2.5 Visualisasi *Artificial Neural Network* (Rayed Bin Wahed, 2016).

Jika kita membahas mengenai ANN, terdapat dua fungsi penting yaitu *basis function* dan *activation function*. *Basis function* adalah fungsi yang memproses *input* yang masuk ke *hidden layer*. Maka untuk *shallow ANN*, *output* dihasilkan dengan fungsi berikut:

$$y(x, w) = f\left(\sum_{j=1}^M w_j \Phi_j (x)\right) \quad (2.1)$$

Di sini, f diganti dengan fungsi aktivasi nonlinier, dan $\Phi_j (x)$ adalah fungsi basis yang disesuaikan dengan koefisien w_j yang diberikan melalui vector (w).

Fungsi aktivasi adalah fungsi yang mengaktifkan neuron untuk melanjutkan *input* ke layer selanjutnya. Secara umum, fungsi aktivasi ini dapat berupa fungsi *nonlinear sigmoidal*.

Adapun fungsi aktivasi secara umum yaitu *logistic sigmoid* dengan persamaan :

$$\sigma(a) = \frac{1}{1+e^{-a}} \quad (2.2)$$

Adapun untuk masalah *multi-class*, yang digunakan adalah fungsi *softmax activation* dengan persamaan :

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (2.3)$$

Adapun fungsi aktivasi lain seperti *hyperbolic tan function* :

$$f(x) = \tanh(x) = \frac{2}{1+e^{-a}} - 1 \quad (2.4)$$

Rectified Linear Units atau ReLU:

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (2.5)$$

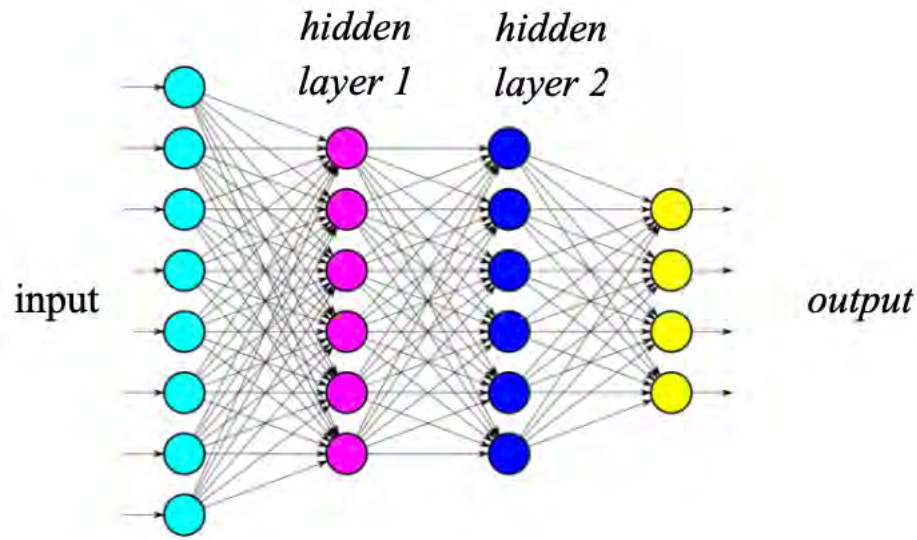
Exponential Linear Units atau ELU:

$$f(x) = \begin{cases} a(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (2.6)$$

(Rayed Bin Wahed, 2016)

2.5. *Deep Neural Network*

Deep Neural Network (DNN) merupakan artificial neural network yang memiliki banyak layer. Umumnya DNN memiliki lebih dari tiga layer yaitu *input* layer, *N hidden layers* dan *output* layer. Proses pembelajaran pada DNN disebut *deep learning*. Secara visual kita dapat menggambarkan *deep neural network* seperti pada Gambar 2.6.



Gambar 2.6 Visualisasi *Deep Neural Network* (Putra, 2019)

Gambar 2.6 merupakan DNN dengan lima layer, sehingga *output* terakhir nya dapat dihitung dengan persamaan berikut :

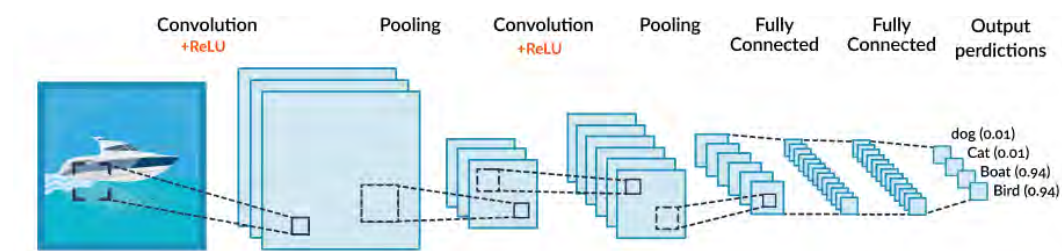
$$\left(\sum_{j=1}^{H_2} u_j, i^{\sigma} \left(\sum_{k=1}^{H_1} v_k, i^{\sigma} \left(\sum_{m=1}^M x_m w_{m,k} + \beta_k \right) + \gamma_i \right) + \lambda_i \right) \quad (2.7)$$

dimana β, γ, λ adalah *noise* atau bias (Putra, 2019).

2.6. *Convolutional Neural Network*

Convolutional neural network (CNN) merupakan arsitektur yang mampu mengenali informasi prediktif suatu objek seperti gambar, teks, potongan suara, dan lain sebagainya. CNN merupakan pengembangan dari *multilayer perceptron* (MLP) yang didesain untuk mengolah data dalam bentuk citra. CNN termasuk dalam jenis *Deep neural network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Penelitian tentang CNN pertama kali dilakukan oleh Hubel dan Wiesel (1968) tentang *visual cortex* pada indera penglihatan kucing. Arsitektur CNN terdiri atas beberapa layer yaitu *convolution*

layer, fungsi *activation layer*, *pooling layer*, dan *fully connected layer*. Ilustrasi jaringan arsitektur Convolutional Neural Network dapat dilihat pada Gambar 2.7.



Gambar 2.7 Ilustrasi Arsitektur *Convolutional Neural Network* (CNN)
(Choudari, 2019)

Tahap pertama dalam arsitektur CNN adalah tahap konvolusi. Kemudian dilanjutkan fungsi aktivasi menggunakan fungsi aktivasi ReLu (*Rectifier Linear Unit*), kemudian dilanjutkan dengan proses *pooling*. Proses ini diulang terus menerus sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke *fully connected neural network*, sehingga dapat dihasilkan *output class*. Penjelasan secara detail dari tahap arsitektur CNN tersebut dijelaskan sebagai berikut.

2.6.1. *Convolution Layer*

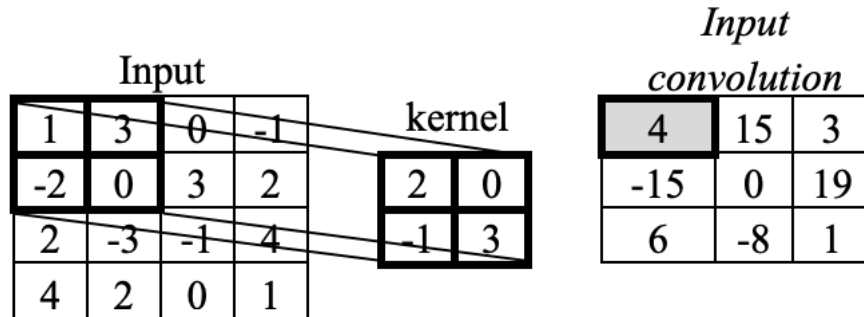
Convolution layer merupakan proses utama yang mendasari jaringan arsitektur CNN dan terdiri atas kernel. Kernel-kernel pada lapisan ini disebut filter konvolusi. Kernel berfungsi mempelajari fitur-fitur lokal pada feature map. Tahap *convolutional layer* melakukan operasi konvolusi pada *output* dari *layer* sebelumnya. Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada output fungsi lain secara berulang. Persamaan konvolusi merupakan persamaan pada dua fungsi argument bernilai riil. Operasi konvolusi $s(t)$ dapat ditunjukkan pada persamaan berikut:

$$s(t) = \sum_a I(a).K(t - a) \quad (2.8)$$

Dimana $I(a)$ adalah input dan $K(a)$ adalah kernel. *Input convolution layer* merupakan gambar yang direpresentasikan menjadi sebuah matriks. Operasi konvolusi menghasilkan nilai tinggi dan rendah pada posisi tertentu pada *feature map*. Posisi tertentu dari konvolusi kernel, merupakan perkalian untuk setiap nilai pada sel kernel dan nilai piksel gambar yang tumpang tindih dengan sel kernel (Khan *et al.*, 2018). Adapun operasi perkaliannya menggunakan persamaan sebagai berikut:

$$h_{i,j} = \sum_{k=1}^m \sum_{l=1}^m W_{k,l} X_{i+k-1,j+l-1} \quad (2.9)$$

Dimana m adalah lebar dan tinggi kernel, h adalah *input convolution*, x adalah input, dan w adalah *convolutional kernel*. Sebagai contoh, ilustrasi pada proses *convolution layer* dengan nilai stride 2 dapat dilihat pada Gambar 2.8.



Gambar 2.8 Ilustrasi Proses *Convolutional Layer* (Khan *et al.*, 2018)

Adapun perhitungan nilai pada *input convolution* yang diperoleh dengan menggunakan persamaan 2.9 adalah sebagai berikut:

$$h_{1,1} = (1.2) + (3.0) + (-2. -1) + (0.3) = 4$$

$$h_{1,2} = (3.2) + (0.0) + (0. -1) + (3.3) = 15$$

$$h_{1,3} = (0.2) + (-1.0) + (3. -1) + (2.3) = 3$$

$$h_{2,1} = (-2.2) + (0.0) + (2. -1) + (-3.3) = -15$$

$$h_{2,2} = (0.2) + (3.0) + (-3. -1) + (-1.3) = 0$$

$$h_{2,3} = (3.2) + (2.0) + (-1. -1) + (4.3) = 19$$

$$h_{3,1} = (2.2) + (-3.0) + (4. -1) + (2.3) = 6$$

$$h_{3,2} = (-3.2) + (-1.0) + (2. -1) + (0.3) = -8$$

$$h_{3,3} = (-1.2) + (4.0) + (0. -1) + (1.3) = 1$$

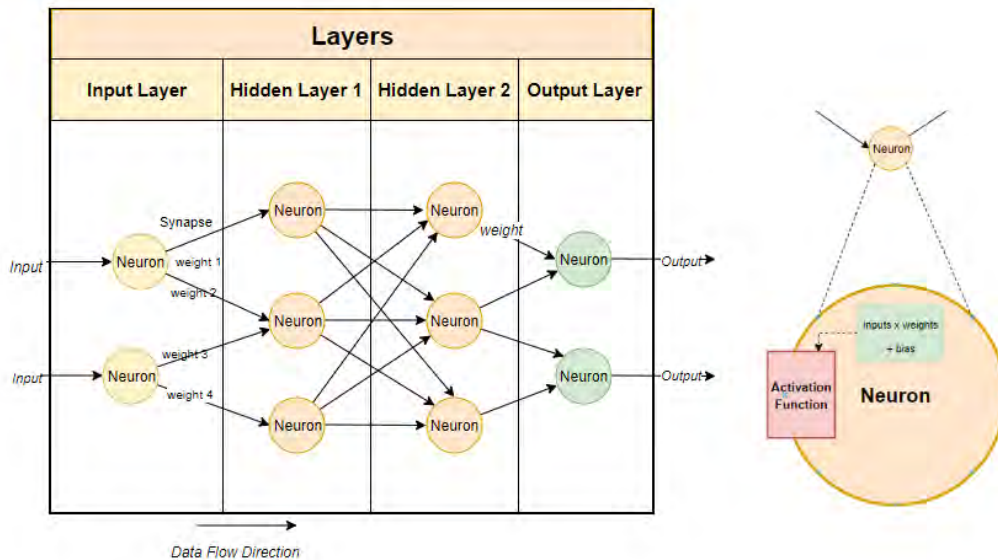
Dari perhitungan diatas, di peroleh matriks *input convolution* sebagai berikut :

$$\begin{bmatrix} 4 & 15 & 3 \\ -15 & 0 & 19 \\ 6 & -8 & 1 \end{bmatrix}$$

(Khan *et al.*, 2018).

2.6.2. *Weights dan Bias*

Ketika *input* ditransmisikan antar *neuron*, *weights* diterapkan dan diteruskan ke fungsi aktivasi bersama dengan bias. Untuk lebih jelasnya perhatikan ilustrasi *weights* dan *bias* pada Gambar 2.9.



Gambar 2.9 Ilustrasi *Weights* dan *Bias* pada *Layer* (Malik, 2019)

Weights mengontrol sinyal (atau kekuatan koneksi) antara dua neuron. Dengan kata lain, bobot menentukan seberapa besar pengaruh input terhadap output. Saat jaringan neural dilatih pada set pelatihan, jaringan akan diinisialisasi dengan satu set *weights*. *Weights* kemudian dioptimasi selama periode pelatihan untuk menghasilkan *weights* yang optimal.

Bias adalah nilai konstanta yang ditambahkan ke layer *input* dengan *weights*. *Bias* digunakan untuk mengimbangi hasil dan untuk menggeser hasil fungsi aktivasi ke nilai positif atau negatif.

Dalam proses perhitungan nya, neuron akan menghitung *weighted sum* dari *input* dengan persamaan:

$$Y = \sum(\text{weight} * \text{input}) + \text{bias} \quad (2.10)$$

Dimana *input* merupakan :

$$x_1, x_2, \dots, x_n \quad (2.11)$$

Dan *weight* adalah :

$$w_1, w_2, \dots, w_n \quad (2.12)$$

Maka *weighted sum* dapat dihitung dengan persamaan :

$$x_1 w_1 + x_2 w_2 + \dots + x_n w_n + \text{bias} \quad (2.13)$$

2.6.3. Fungsi Aktivasi

Fungsi aktivasi merupakan operasi matematik yang dikenakan pada sinyal output y . Fungsi aktivasi berfungsi menentukan apakah suatu *neuron* aktif atau tidak berdasarkan *weighter sum* dari suatu *input*. Beberapa jenis fungsi aktivasi yang sering digunakan pada *deep learning* adalah *sigmoid*, *Tanh*, *algebraic*

sigmoid, ReLU, noisy ReLU, Leaky ReLU/PReLU, Randomized Leaky ReLU, dan *Eksponential Linear Unit*. Penelitian ini menggunakan dua fungsi aktivasi yaitu fungsi aktivasi ReLU dan fungsi aktivasi *softmax*. Penjelasan dari kedua fungsi tersebut adalah sebagai berikut.

a. ReLU

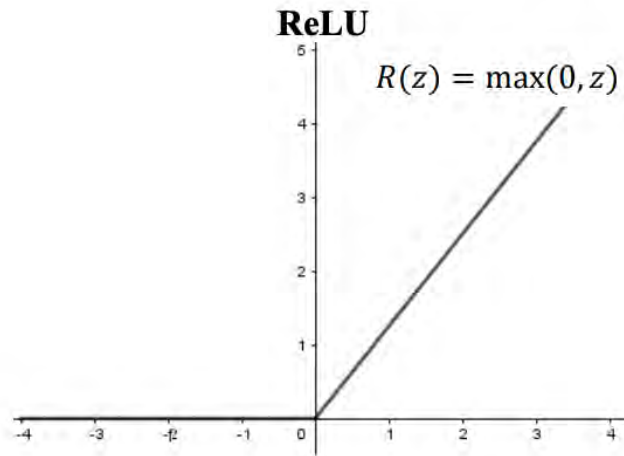
Fungsi aktivasi *Rectifier Linear Unit*(ReLU) merupakan fungsi aktivasi sederhana yang memiliki kepentingan praktis khusus karena perhitungannya yang cepat. Kelebihan fungsi aktivasi ReLU dibandingkan dengan fungsi aktivasi lain adalah sebagai berikut.

1. Fungsi aktivasi ReLU merupakan fungsi aktivasi default ketika mengembangkan *multilayer perceptron* dan *convolutional neural network*.
2. Fungsi aktivasi ReLU mengatasi masalah *gradient descent* yang hilang, yang memungkinkan model belajar lebih cepat dan berkinerja lebih baik.
3. Menemukan cara melatih jaringan dengan lebih cepat, sehingga mengurangi kemungkinan terjadinya *overfitting*.

Fungsi aktivasi ReLU memetakan input ke 0 jika negatif dan mempertahankan nilainya jika positif. Representasi fungsi ReLU adalah sebagai berikut

$$f_{ReLU}(x) = \max(0, x) \quad (2.14)$$

Adapun grafik nya dapat dilihat pada Gambar 2.10.



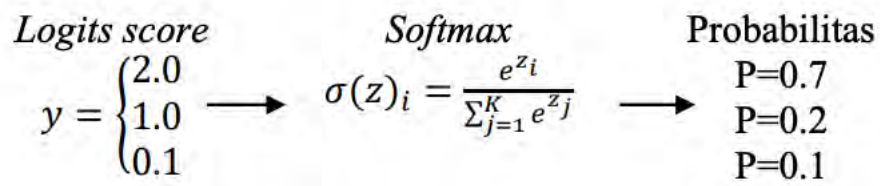
Gambar 2.10 Grafik Fungsi Aktivasi ReLU (Khan *et al.*, 2018)

b. Fungsi Aktivasi *Softmax*

Fungsi aktivasi *softmax* merupakan fungsi input vektor dari bilangan real K , yang kemudian dinormalkan menjadi distribusi probabilitas yang terdiri atas probabilitas K yang proporsional ke eksponensial input. Komponen vektor pada *softmax* memiliki interval $(0,1)$. Fungsi *softmax* merupakan lapisan yang menghubungkan antara *fully connected layer* dengan *dense connection*. *Softmax* berfungsi untuk menghitung probabilitas pada setiap kelas target yang memungkinkan dan akan membantu menentukan kelas target pada input yang diberikan. Nilai *softmax* berada pada rentang probabilitas output dari 0 hingga 1 dan jumlah semua probabilitas sama dengan satu. Definisi fungsi *softmax* $\sigma = \mathbb{R}^K \rightarrow \mathbb{R}^K$ dituliskan sebagai persamaan berikut:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.15)$$

dimana $i = 1, \dots, K$ dan $z = (z_1, \dots, z_K) \in \mathbb{R}^K$. Adapun contoh ilustrasi fungsi aktivasi *softmax* dapat dilihat pada Gambar 2.11 sebagai berikut.

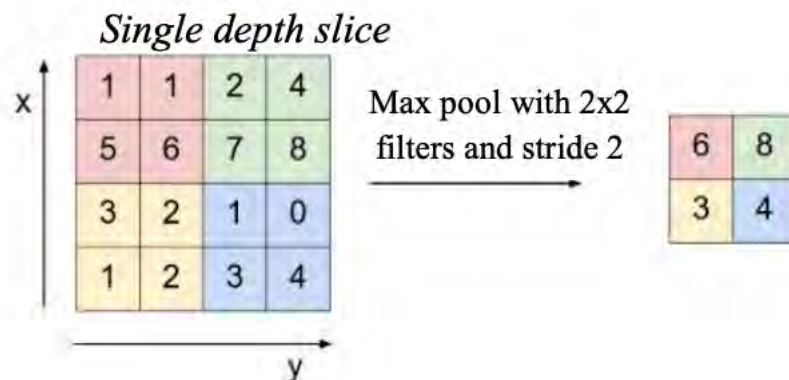


Gambar 2.11 ilustrasi fungsi aktivasi *softmax*

Logits score menunjukkan lapisan *neuron* terakhir sebagai *output* mentah pada lapisan terakhir neural network sebelum proses aktivasi berlangsung. Setelah *output* diproses dengan *softmax* akan menghasilkan nilai probabilitas dengan jumlah 1 (Khan *et al.*, 2018).

2.6.4. *Pooling Layer*

Pooling layer terletak setelah *convolution layer*. *Pooling layer* terdiri atas sebuah *filter* dengan ukuran dan *stride* tertentu yang secara bergantian bergeser pada seluruh *area feature map*. Jenis *pooling layer* yang biasa digunakan yaitu *average pooling* dan *max pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata dan pada *max pooling* yang diambil adalah nilai maksimal. Lapisan *pooling* yang dimasukkan diantara lapisan konvolusi pada model CNN dapat mengurangi ukuran *volume output* pada *feature map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan. Adapun proses dari *max pooling* dapat dilihat pada Gambar 2.12.



Gambar 2.12 Operasi *max pooling* (Khan *et al.*, 2018)

Output dari proses *pooling* adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan citra awal. *Pooling layer* beroperasi pada blok *input feature map* dan menggabungkan aktivasi fitur. Operasi kombinasi antara *pooling layer* dan fungsi aktivasi didefinisikan sebagai fungsi penyatuan seperti fungsi rata-rata atau maksimal.

2.6.5. *Fully Connected Layer*

Fully connected layer merupakan sebuah lapisan dimana semua *neuron* aktivasi dari lapisan sebelumnya terhubung dengan *neuron* lapisan selanjutnya. Perbedaan *fully connected layer* dengan konvolusi biasa adalah *neuron* pada lapisan konvolusi terhubung hanya ke daerah tertentu sedangkan *fully connected* memiliki *neuron* yang semuanya terhubung (Khan *et al.*, 2018).

2.7. *Inception V3*

Inception V3 berfokus pada penggunaan daya komputasi yang lebih sedikit dengan memodifikasi arsitektur Inception sebelumnya. Ide ini diusulkan dalam makalah *Rethinking the Inception Architecture for Computer Vision*, yang diterbitkan pada tahun 2015. Ide ini disusun bersama oleh *Christian Szegedy*,

Vincent Vanhoucke, Sergey Ioffe, dan Jonathon Shlens. Dibandingkan dengan *VGGNet, Inception Networks (GoogLeNet / Inception v1)* terbukti lebih efisien secara komputasi, baik dari jumlah parameter yang dihasilkan oleh jaringan maupun *cost* memori dan sumber daya lainnya. Jika ada perubahan yang ingin dilakukan pada *Inception Network*, harus dilakukan secara hati-hati untuk memastikan bahwa keunggulan komputasi tidak hilang. Dengan demikian, adaptasi jaringan *Inception* untuk berbagai kasus berbeda ternyata menjadi masalah karena ketidakpastian efisiensi jaringan baru. Dalam model *Inception v3*, beberapa teknik untuk mengoptimalkan jaringan telah disarankan untuk mengurangi kendala agar adaptasi model menjadi lebih mudah. Teknik tersebut meliputi *factorized convolutions, regularization, dimension reduction, dan parallelized computations* (Kurama, 2020).

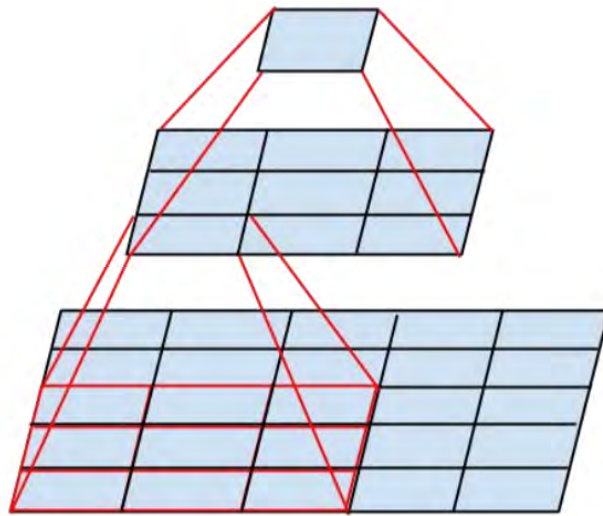
Arsitektur jaringan *Inception v3* dibangun secara progresif, langkah demi langkah, seperti yang dijelaskan di bawah ini:

1. *Factorizing Convolutions*

Tujuan dari faktorisasi konvolusi adalah untuk mengurangi jumlah koneksi atau parameter tanpa mengurangi efisiensi jaringan.

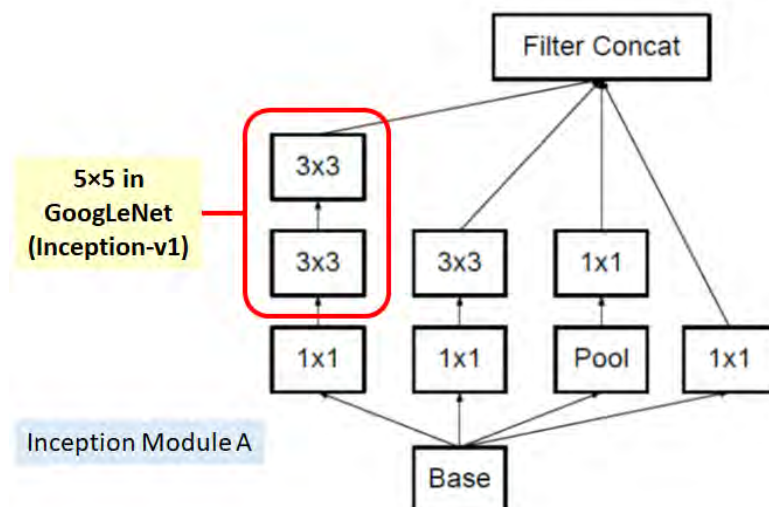
1.1. *Factorization Into Smaller Convolutions*

Dua konvolusi 3x3 digantikan dengan satu konvolusi 5x5 seperti pada Gambar 2.13



Gambar 2.13 dua konvolusi 3x3 digantikan oleh konvolusi 5x5
(Szegedy *et al.*, 2016)

Dengan menggunakan 1 layer filter 5x5, maka jumlah parameter nya adalah $= 5 \times 5 = 25$, jika menggunakan 2 layer filter 3x3 maka jumlah parameter nya adalah $= 3 \times 3 + 3 \times 3 = 18$, sehingga jumlah parameter nya berkurang sebanyak 28%. Teknik yang sama juga telah diterapkan oleh arsitektur *VGGNet*. Pada InceptionV3, teknik diterapkan pada *Inception Module A*, Ilustrasi dapat dilihat pada Gambar 2.14.

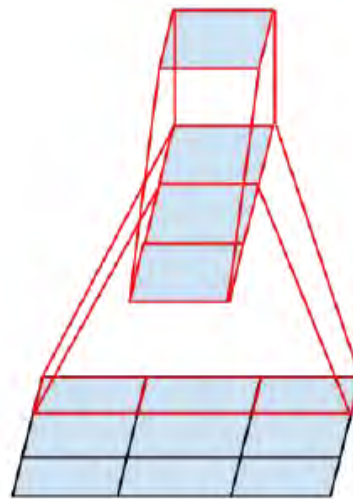


Gambar 2.14 Inception Module A dengan factorization

(Szegedy *et al.*, 2016)

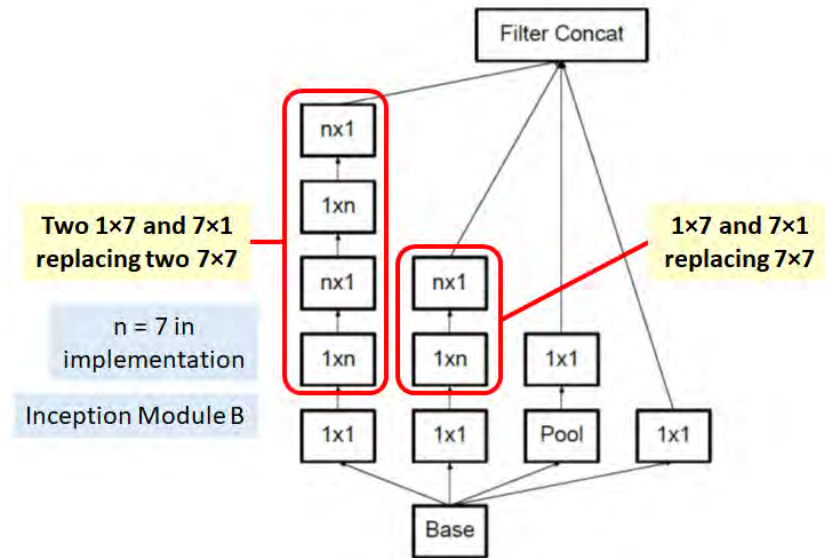
1.2. *Factorization Into Asymmetric Convolutions*

Satu konvolusi 3×1 yang diikuti oleh konvolusi 1×3 digantikan dengan konvolusi 3×3 seperti pada Gambar 2.15.



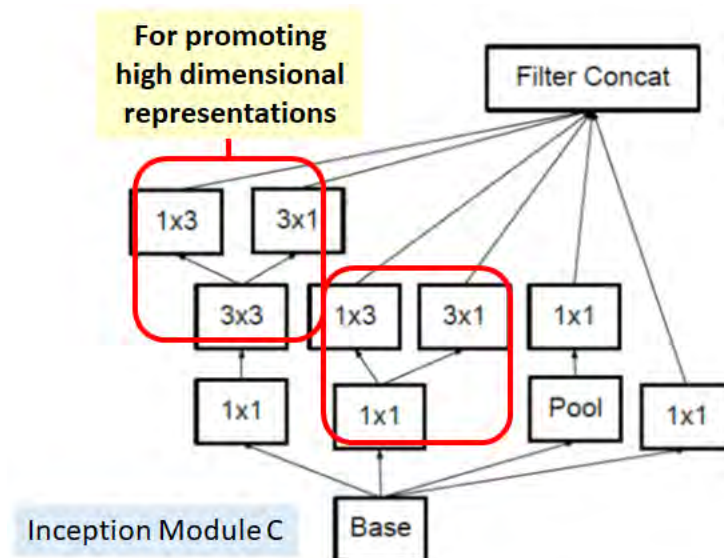
Gambar 2.15 satu konvolusi 3×1 yang diikuti konvolusi 1×3 diganti oleh konvolusi 3×3 (Szegedy *et al.*, 2016)

Dengan menggunakan filter 3×3 , maka jumlah parameter nya adalah $= 3 \times 3 = 9$, jika menggunakan filter 3×1 dan 1×3 , maka jumlah parameternya adalah $= 3 \times 1 + 1 \times 3 = 6$, sehingga jumlah parameter nya berkurang sebanyak 33%. Teknik ini diterapkan pada *Inception Module B*, ilustrasi dapat dilihat pada Gambar 2.16



Gambar 2.16 Inception Module B dengan *asymmetric factorization* (Szegedy *et al.*, 2016)

Selanjutnya, *Inception module C* juga digunakan, untuk mempromosikan representasi dimensi tinggi seperti yang ditunjukkan pada Gambar 2.17

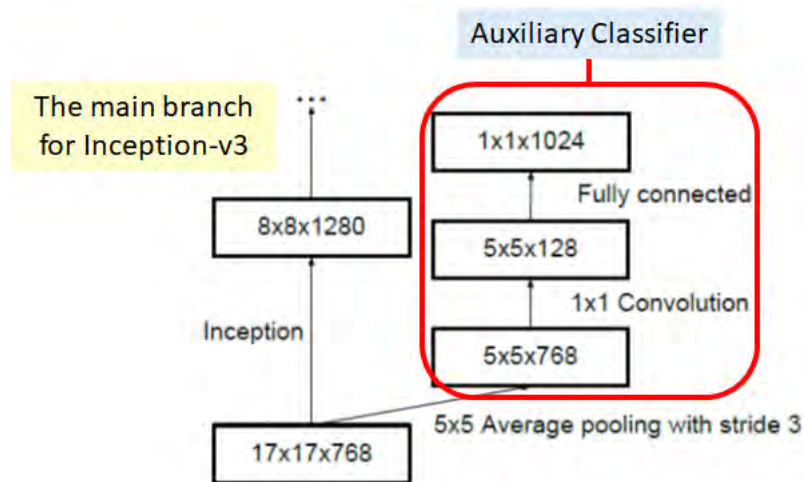


Gambar 2.17 Inception Module C dengan *asymmetric factorization* (Szegedy *et al.*, 2016)

Dengan diterapkannya tiga module inception tadi, jumlah parameter didalam jaringan akan semakin sedikit, sehingga akan memperkecil kemungkinan terjadinya *overfitting* (Tsang, 2018).

2. *Auxiliary Classifier*

Auxiliary classifier sudah diterapkan sejak arsitektur *GoogLeNet/InceptionV1*. Ada sedikit modifikasi yang dilakukan pada *InceptionV3*, hanya satu *auxiliary classifier* yang digunakan sebelum *layer 17x17* terakhir, dibandingkan dengan arsitektur sebelumnya yang menggunakan dua *auxiliary classifier*. Dalam *InceptionV3* *auxiliary classifier* digunakan sebagai *regularizer*, ilustrasinya dapat dilihat pada Gambar 2.18.

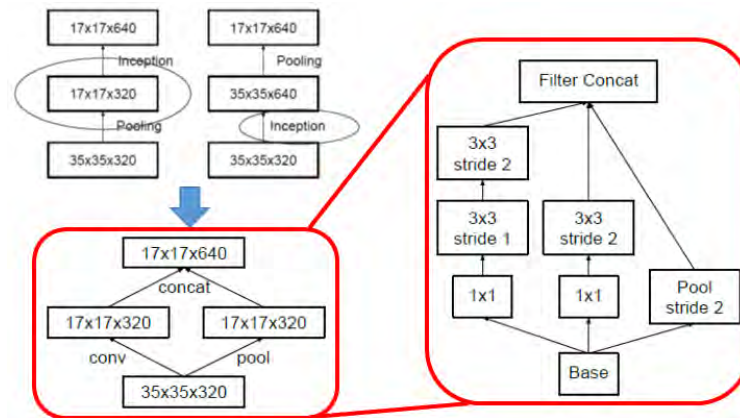


Gambar 2.18 *Auxiliary Classifier* sebagai *regularization* (Szegedy *et al.*, 2016)

3. *Efficient Grid Size Reduction*

Umumnya seperti yang diterapkan pada arsitektur *AlexNet* dan *VGGNet*, pengurangan *feature map* dilakukan oleh *max pooling*. Tetapi *max pooling* ini memiliki kekurangan yaitu terlalu berlebihan dalam mengurangi *feature map* yang diikuti oleh *layer* konvolusi, atau terlalu boros ketika konvolusi

layer yang diikuti oleh *max pooling*. Pada InceptionV3 *efficient grid size reduction* diterapkan seperti ilustrasi pada Gambar 2.19.

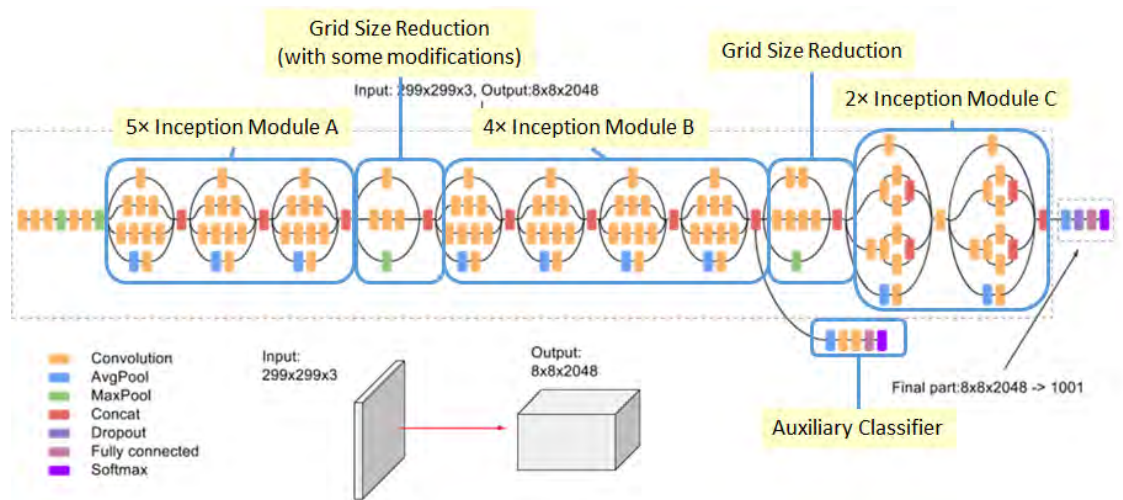


Gambar 2.19 Conventional downsizing (atas kiri), *Efficient Grid Size Reduction* (bawah kiri), Detail dari *Efficient Grid Size Reduction* dalam Arsitektur (kanan) (Szegedy *et al.*, 2016)

Dengan *efficient grid size reduction*, 320 *feature maps* dihasilkan oleh konvolusi dengan *stride 2*. *Feature map* tadi diperoleh dari *max pooling*, kemudian dua set *feature maps* digabungkan menjadi 640 *feature maps* yang kemudian akan diteruskan ke *module inception* selanjutnya (Tsang, 2018).

4. Arsitektur InceptionV3

Semua proses diatas diterapkan dalam arsitektur *InceptionV3*, Ilustrasi arsitektur nya dapat dilihat pada Gambar 2.20.



Gambar 2.20 Arsitektur *InceptionV3* (Tsang, 2018)

Parameter dari arsitektur *InceptionV3* dapat dilihat pada Tabel 2.2.

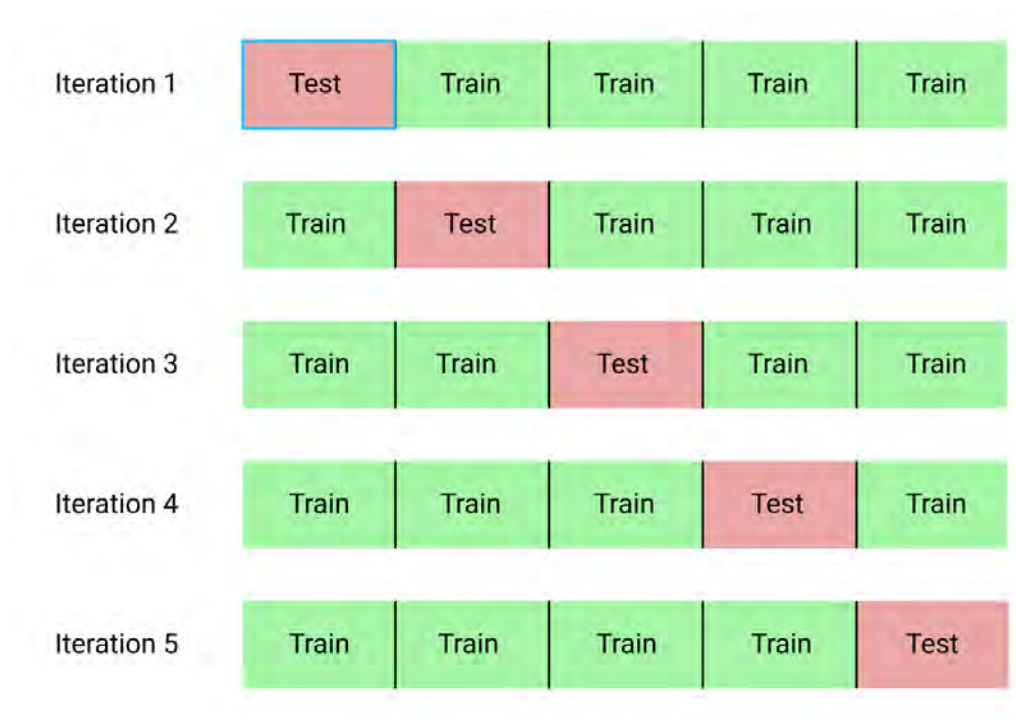
Tabel 2.2. Parameter Arsitektur InceptionV3

<i>Type</i>	Filter Size / Stride	Input Size
Conv	3 x 3 / 2	299 x 299x 3
Conv	3 x 3 / 1	149 x 149 x 32
Conv padded	3 x 3 / 1	147 x 147 x 32
Pool	3 x 3 / 2	147 x 147 x 64
Conv	3 x 3 / 1	73 x 73 x 64
Conv	3 x 3 / 2	71 x 71 x 80
Conv	3 x 3 / 1	35 x 35 x 192
3 x Inception A	Seperti Gambar 2.13	35 x 35 x 288
5 x Inception B	Seperti Gambar 2.15	17 x 17 x 768
2 x Inception C	Seperti Gambar 2.16	8 x 8 x 1280
Pool	8 x 8	8 x 8 x 2048

<i>Type</i>	Filter Size / Stride	Input Size
Linear	Logits	1 x 1 x 2048
Softmax	Classifier	1 x 1 x 1000

2.8. *K-Fold Cross Validation*

K-Fold Cross Validation adalah metode yang melakukan pembagian data secara acak menjadi kelompok k atau lipatan dengan ukuran yang kira-kira sama. Lipatan pertama disimpan untuk *validation* dan *model ditraining* pada lipatan k-1. Proses ini diulang sebanyak k kali dan setiap kali lipatan yang berbeda atau kelompok titik data yang berbeda digunakan untuk validasi. Ilustrasi dapat dilihat pada Gambar 2.21.

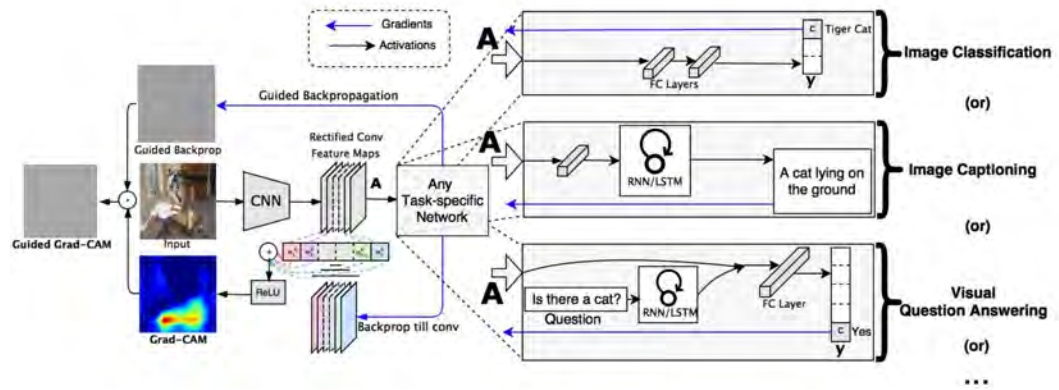


Gambar 2.21 Contoh 5 fold cross validation (Shaikh, 2018).

Seiring berjalannya proses hingga banyak k , kita mendapatkan k kali *mean square error* (MSE). MSE_1, MSE_2, hingga MSE_K, sehingga MSE nya didapatkan melalui rata-rata dari MSE terhadap lipatan k . *mean square error* dari k fold cross validation ini dapat dihitung dengan persamaan berikut (Khandelwal, 2018).

$$cv_k = \frac{1}{k} \sum_{i=1}^k MSE_i \quad (2.16)$$

2.9. Grad-CAM



Gambar 2.22 Visualisasi *Grad-CAM* (Selvaraju et al., 2019).

Grad-CAM menggunakan informasi gradien yang berada pada lapisan konvolusional terakhir CNN untuk menetapkan nilai penting ke seluruh *neuron* untuk setiap keputusan tertentu. Meskipun teknik ini cukup umum karena dapat digunakan untuk menjelaskan *layer* aktivasi manapun dari dalam *network*, *Grad-CAM*, berfokus pada *layer output* saja. Seperti yang ditunjukkan pada Gambar 2.21, untuk mendapatkan kelas diskriminatif peta lokalisasi *Grad-CAM* $L_{\text{Grad-CAM}}^c \in \mathbb{R}^{u \times v}$ dari lebar u dan tinggi v untuk setiap kelas c , pertama kita menghitung gradien skor untuk kelas c , y^c (sebelum *softmax*), sehubungan dengan aktivasi

feature map A^k dari *layer* konvolusi seperti $\frac{\partial y^c}{\partial A^k}$. Gradien yang mengalir kembali ini adalah *global average pooled* pada dimensi lebar dan tinggi (masing-masing di indeks oleh i dan j) untuk mendapatkan bobot *neuron* α_k^c yang dapat dituliskan pada persamaan berikut :

$$\alpha_k^c = \overbrace{\frac{1}{Z} \sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}} \quad (2.17)$$

Selama proses komputasi α_k^c , ketika gradien *backpropagating* sehubungan dengan aktivasi, jumlah komputasi yang tepat untuk produk maktriiks berturut-turut dari matriks bobot dan gradien sehubungan dengan fungsi aktivasi hingga akhir konvolusi dimana gradien disebarkan. Oleh karena itu, bobot α_k^c ini merepresentasikan linierisasi parsial *deep network* di hilir dari A , dan menangkap 'pentingnya' *feature map* k untuk kelas target c . Grad-CAM melakukan kombinasi *forward activation map*, dan dilanjutkan dengan ReLU untuk mendapatkan :

$$L_{\text{Grad-CAM}}^c = \text{ReLU} \left(\underbrace{\sum_k \alpha_k^c A^k}_{\text{linear combination}} \right) \quad (2.18)$$

ReLU diterapkan ke kombinasi linier *maps* karena Grad-CAM hanya tertarik pada fitur yang memiliki pengaruh positif pada *class of interest*, misalnya piksel yang intensitasnya harus ditingkatkan untuk meningkatkan y^c . Piksel negatif cenderung termasuk kategori lain dalam gambar. Seperti yang diduga, tanpa ReLU, *localization maps* terkadang menyorot lebih dari sekadar kelas yang diinginkan dan berperforma lebih buruk dalam *localization*. Secara umum, y^c tidak perlu berupa skor kelas yang dihasilkan oleh klasifikasi gambar CNN. (Selvaraju *et al.*, 2019).

2.10. Data Augmentation

Data Augmentation merupakan teknik yang digunakan untuk meningkatkan jumlah data dengan menambahkan salinan yang sedikit dimodifikasi dari data yang sudah ada atau data sintetis yang baru dibuat dari data yang ada. *Image Data Augmenter* mengkonfigurasi serangkaian opsi *preprocessing* untuk augmentasi gambar seperti perubahan ukuran, rotasi dan refleksi. Berikut penjelasan dari setiap opsi nya:

1. FillValue

Digunakan untuk menentukan titik di luar batas saat pengambilan sampel ulang, ditentukan sebagai skalar numerik atau vektor numerik.

2. RandXReflection

Refleksi acak di arah kiri-kanan, ditentukan sebagai skalar logis.

3. RandYReflection

Refleksi acak dalam arah atas-bawah, ditetapkan sebagai skalar logis.

4. RandRotation

Rentang rotasi dalam derajat yang diterapkan ke gambar masukan.

5. RandScale

Rentang skala seragam (isotropik) yang diterapkan pada gambar masukan.

6. RandXScale

Rentang skala horizontal diterapkan ke gambar *input*.

7. RandYScale

Rentang skala vertikal diterapkan ke gambar masukan.

8. *RandXShear*

Rentang pemotongan horizontal yang diterapkan pada citra masukan.

9. *RandYShear*

Rentang pemotongan vertikal yang diterapkan pada citra masukan.

10. *RandXTranslation*

Rentang pergeseran horizontal yang diterapkan ke gambar masukan.

11. *RandYTranslation*

Rentang pergeseran vertikal yang diterapkan ke gambar masukan

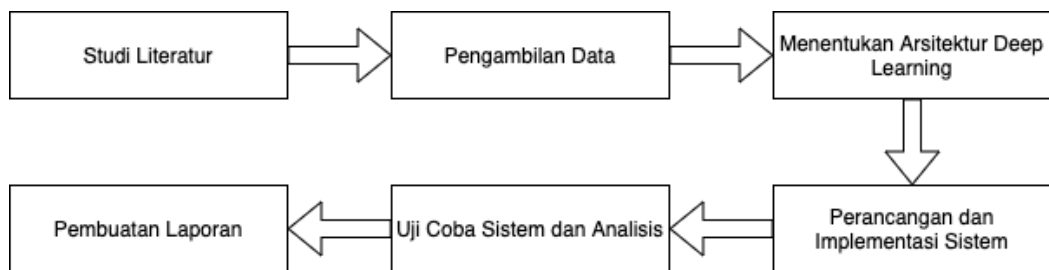
(Beale, Hagan and Demuth, 2020).

BAB III

METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Tahapan penelitian yang dilakukan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Tahapan Penelitian

Adapun uraian tahapan penelitian pada *Gambar 3.1* sebagai berikut.

1. Studi Literatur dilakukan untuk mencari referensi teori yang relevan terkait topik penelitian yang dilakukan yaitu mengenai metode untuk klasifikasi bakteri. Tahap ini adalah tahap awal penelitian ini dengan mencari sejumlah referensi dari jurnal nasional maupun internasional, dan laporan penelitian.
2. Pengambilan *dataset* bakteri diambil melalui sumber *DIBaS* (<http://misztal.edu.pl/software/databases/dibas/>).
3. Menentukan arsitektur *deep learning* yang digunakan yaitu menggunakan arsitektur *InceptionV3*.
4. Perancangan sistem dilakukan dengan pembuatan *flowchart* terkait alur kerja sistem.

5. Implementasi sistem dilakukan sesuai dengan rancangan *flowchart*. Sistem dibuat menggunakan Matlab 2020b dengan *add-on Deep learning toolbox*.
6. Uji coba sistem dilakukan untuk mengetahui akurasi dari sistem yang telah dibuat.
7. Tahap akhir dari penelitian ini adalah melakukan penulisan laporan penelitian dalam bentuk skripsi sebagai bahan publikasi.

3.2. Waktu dan Lokasi Penelitian

Penelitian dilakukan selama 10 bulan yang dimulai sejak bulan Januari 2020 hingga proses penulisan laporan penelitian pada bulan Oktober 2020. Penelitian ini dilakukan di laboratorium Kecerdasan Buatan Departemen Teknik Informatika Universitas Hasanuddin Gowa.

3.3. Instrumen Penelitian

Instrumen penelitian yang digunakan pada penelitian ini meliputi :

1. Software
 - MacOS 10.14.6
 - Matlab R2020b (9.9.0.1467703)
 - Matlab Add-On Deep Learning Toolbox
2. Hardware
 - MacBook Pro (Retina, 15-inch, Mid 2015), Processor 2,2 GHz Intel Core i7

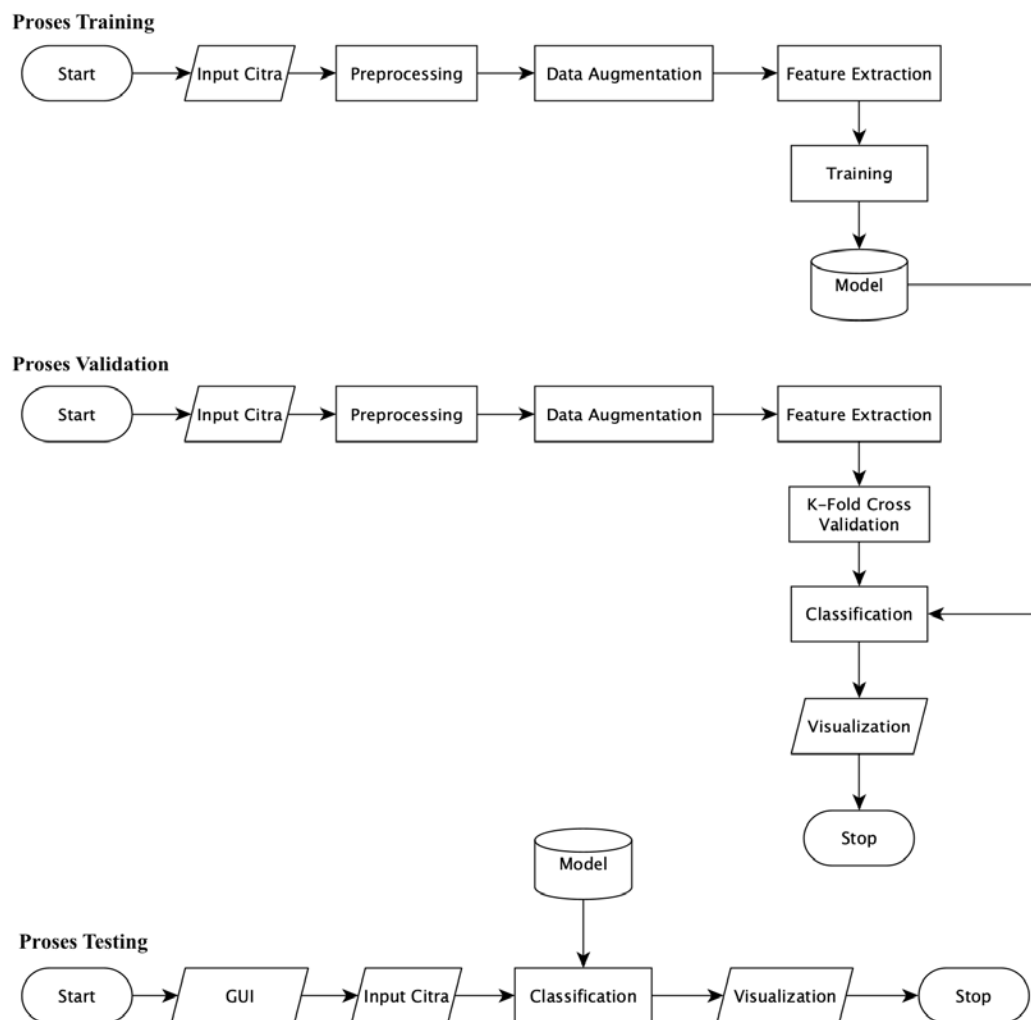
3.4. Teknik Pengambilan Data

Data yang diambil berupa citra dengan resolusi 2048 x 1532 sebanyak 20 gambar tiap kelas bakteri dengan total 108 gambar yang nantinya akan dibagi

menjadi data *training* dan *validation*. Data citra bakteri diperoleh melalui sumber *DIBaS* (<http://misztal.edu.pl/software/databases/dibas/>).

3.5. Perancangan Sistem

Dalam perancangan sistem untuk klasifikasi bakteri ini terbagi menjadi tiga bagian, yaitu *training*, *validation*, dan *testing* seperti yang ditunjukkan Gambar 3.2.



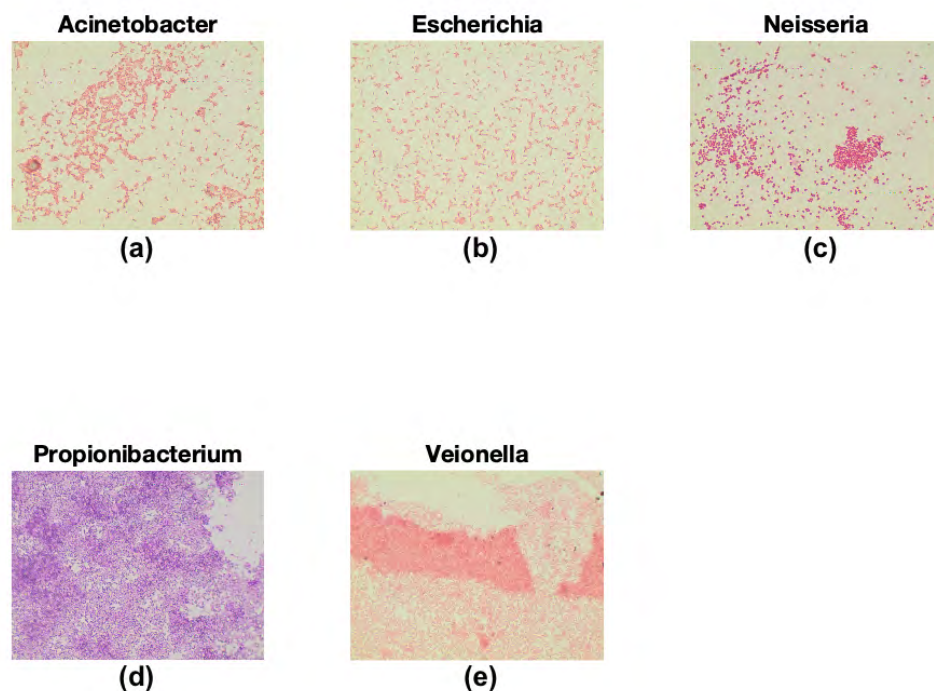
Gambar 3.2 *Flowchart* Perancangan Sistem

Berdasarkan Gambar 3.2 rancangan sistem dapat diuraikan sebagai berikut.

3.5.1. Proses *Training*

1. *Input Citra*

Data yang digunakan untuk proses *training* diambil melalui (<http://misztal.edu.pl/software/databases/dibas/>). Data citra bakteri ini merupakan bakteri yang telah melalui proses pewarnaan gram dan diambil dengan mikroskop Olympus CX31 yang dilengkapi kamera SC30. Data yang berjumlah 108 buah gambar ini terdiri dari lima kelas bakteri yaitu *Acinetobacter baumannii*, *Escherichia coli*, *Neisseria gonorrhoeae*, *Propionibacterium acnes* dan *Veionella*. Contoh data *training* dari setiap kelas bakteri, dapat dilihat pada Gambar 3.3.



Gambar 3.3 Contoh data latih : (a) *Acinetobacter baumannii*, (b) *Escherichia coli*, (c) *Neisseria gonorrhoeae*, (d) *Propionibacterium acnes*, (e) *Veionella*.

Ciri-ciri atau *feature* dari setiap kelas bakteri yang digunakan pada data *training* adalah sebagai berikut:

a. *Acinetobacter*

Memiliki bentuk *rod shaped*, atau masuk kategori *bacilli* dengan bentuk *coccobacillus* (bentuk elips), dan memiliki warna *gram-negative* (pink kemerahan).

b. *Escherichia*

Bentuknya masuk ke kategori *bacilli* dengan bentuk *bacillus* (batang), dan memiliki warna *gram-negative* (pink kemerahan).

c. *Neisseria*

Bentuknya masuk ke kategori *cocci* dengan bentuk *diplococci* (lingkaran sepasang), dan memiliki warna *gram-negative* (pink kemerahan).

d. *Propionibacterium*

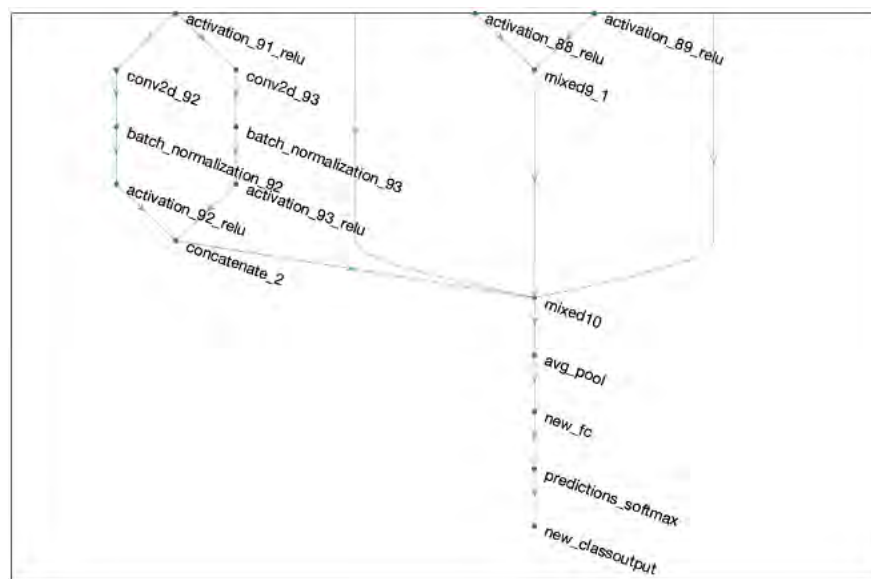
Bentuknya masuk ke kategori *bacilli* dengan bentuk *bacillus* (batang), dan memiliki warna *gram-positive* (violet).

e. *Veionella* masuk ke kategori *cocci* dengan bentuk *diplococci* (lingkaran sepasang), dan memiliki warna *gram-negative* (pink kemerahan)

2. *Preprocessing*

Sebelum *pretrained network InceptionV3* digunakan, pertama-tama kedua *layer* bagian terakhir pada arsitektur, yaitu *layer* “predictions” dan *layer* “*ClassificationLayer_predictions*” akan disesuaikan terlebih dahulu agar dapat mengklasifikasikan dataset yang digunakan yaitu dataset yang terdiri dari lima

jenis kelas bakteri. Sebelum dilakukan proses modifikasi *layer*, pertama-tama kedua *layer* yang akan diubah dicari terlebih dahulu dengan menggunakan fungsi “*findLayersToReplace*”. Setelah *layer* ditemukan, selanjutnya *layer* “*predictions*” dimodifikasi untuk disesuaikan dengan jumlah kelas dataset yang digunakan yaitu lima jenis kelas bakteri. Selanjutnya, *Layer* “*ClassificationLayer_predictions*” ini masih berisi *class* label dari *pretrained network*, sehingga akan digantikan dengan *layer* “*new_classoutput*”, yang tidak memiliki *class* label. Hasil penyesuaian *layer* pada arsitektur InceptionV3 dapat dilihat pada Gambar 3.4.



Gambar 3.4 Hasil Penyesuaian *Layer InceptionV3*

Dapat dilihat pada Gambar 3.4, kedua *layer* paling terakhir yang berada pada bagian bawah, telah disesuaikan agar dapat mengklasifikasikan lima jenis bakteri dan diubah namanya menjadi menjadi “*predictions_softmax*” dan “*new_classoutput*”.

Berikut potongan *source code* untuk fungsi *findLayersToReplace* :

```
% Get source, destination, and layer names.
src = string(lgraph.Connections.Source);
dst = string(lgraph.Connections.Destination);
layerNames = string({lgraph.Layers.Name}');

% Find the classification layer. The layer graph must have a
single
% classification layer.
isClassificationLayer = arrayfun(@(l) ...

(isa(l,'nnet.cnn.layer.ClassificationOutputLayer')|isa(l,'nnet.
layer.ClassificationLayer')), ...
    lgraph.Layers);

if sum(isClassificationLayer) ~= 1
    error('Layer graph must have a single classification
layer.')
end
classLayer = lgraph.Layers(isClassificationLayer);
```

Berikut *source code* untuk modifikasi *layer* :

```
%input size
inputSize = net.Layers(1).InputSize;
%fully connected
if
isa(learnableLayer,'nnet.cnn.layer.FullyConnectedLayer')
    newLearnableLayer = fullyConnectedLayer(numClasses,
...
        'Name','new_fc', ...
        'WeightLearnRateFactor',15, ...
        'BiasLearnRateFactor',15);

%output
newClassLayer =
classificationLayer('Name','new_classoutput');
lgraph =
replaceLayer(lgraph,classLayer.Name,newClassLayer);
```

Data citra selanjutnya akan melalui proses *labelling* yang dilakukan untuk membagi data menjadi lima kelas yang nantinya akan digunakan pada proses *training* dan *validation*, jumlah data beserta *label* nya dapat dilihat pada Gambar 3.5. Setelah itu, dilakukan proses pembagian data *training* dan *validation* dengan metode *k-fold cross validation* dengan jumlah $k=5$. Hasil

pembagian data ini dapat dilihat pada Gambar 3.6. *K-fold cross validation* dilakukan agar setiap citra didalam *dataset* memiliki peluang untuk menjadi data *training* dan *validation*, sehingga nantinya kita mendapatkan *dataset* terbaik untuk menghasilkan akurasi tertinggi.

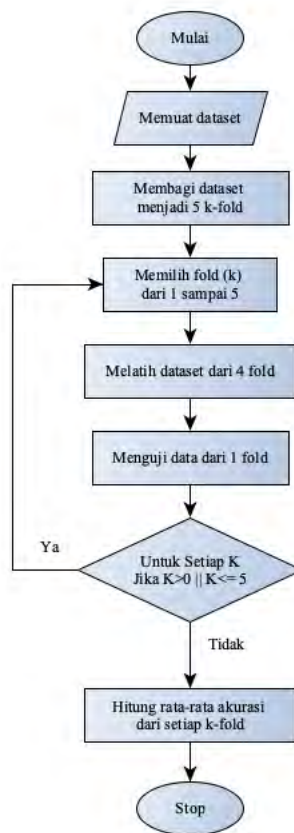
	Label	Count
1	Acinetobacter	20
2	Escherichia	20
3	Neisseria	23
4	Propionibac...	23
5	Veionella	22

Gambar 3.5 Jumlah *dataset* setiap kelas

```
i = 1
Numberoftraindata = 86
Numberofvalidationdata = 22
i = 2
Numberoftraindata = 87
Numberofvalidationdata = 21
i = 3
Numberoftraindata = 86
Numberofvalidationdata = 22
i = 4
Numberoftraindata = 86
Numberofvalidationdata = 22
i = 5
Numberoftraindata = 87
Numberofvalidationdata = 21
```

Gambar 3.6 Pembagian data *training* dan *validation* menggunakan *k-fold cross validation*

Adapun flowchart dari proses *k-fold cross validation* untuk *source code* dibawah, dapat dilihat pada Gambar 3.7.



Gambar 3.7 Flowchart Proses *K-fold Cross Validation*

Berikut *source code* untuk proses *k-fold cross validation*:

```
k = 5;
total = 0;
idx = crossvalind('Kfold', k, k);
[imd1,imd2,imd3,imd4,imd5] =
splitEachLabel(imds,0.2,0.2,0.2,0.2,'randomize');

partStores{1} = imd1.Files ;
partStores{2} = imd2.Files ;
partStores{3} = imd3.Files ;
partStores{4} = imd4.Files ;
partStores{5} = imd5.Files ;

for i = 1 :k
    i
    test_idx = (idx == i);
    train_idx = ~test_idx;

    imdsValidation =
imageDatastore(partStores{test_idx},
'IncludeSubfolders', true, 'LabelSource',
'folderNames');

    imdsTrain = imageDatastore(cat(1,
partStores{train_idx}), 'IncludeSubfolders', true,
'LabelSource', 'folderNames');

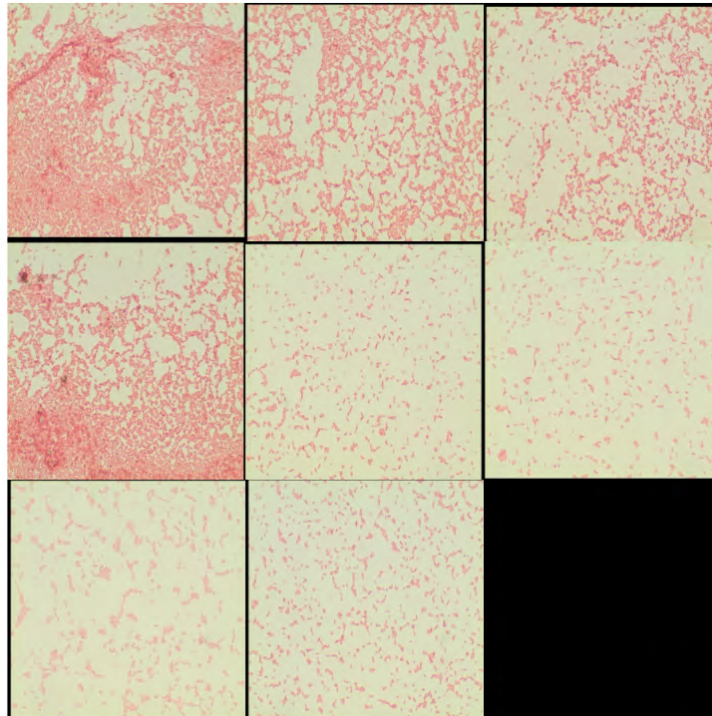
%jumlah data training
Numberoftraindata = numel(imdsTrain.Files)
%jumlah data validasi
Numberofvalidationdata = numel(imdsValidation.Files)
```

3. *Data Augmentation*

Data augmentation memberikan lebih banyak variasi data dalam *network* dengan cara melakukan berbagai macam proses transformasi secara random seperti *RandXReflection*, *RandXTranslation*, dan *RandYTranslation* pada setiap data *training* dan memberikannya ke *network*, sehingga nantinya *network* berisi berbagai macam variasi dataset. teknik *data augmentation* yang digunakan

pada penelitian ini menggunakan teknik *offline augmentation*. *Offline augmentation* ini digunakan untuk menambahkan jumlah dari *dataset* yang sedikit, dengan cara menghasilkan *data training* yang lebih banyak dan tidak membutuhkan *storage* tambahan (Gu, Pednekar and Slater, 2019).

RandXReflection, *RandXTranslation*, dan *RandYTranslation* merupakan transformasi yang digunakan pada penelitian ini karena ketiga transformasi tersebut merupakan transformasi yang umum digunakan dalam *data augmentation* dan akan menghasilkan model yang lebih optimal. Adapun untuk jumlah datanya, *data augmentation* menghasilkan jumlah data berdasarkan konfigurasi *epoch* dan banyak jenis transformasi yang digunakan dalam proses proses *training*. Dalam penelitian ini, digunakan konfigurasi 5 epoch, dengan data *training* sebanyak 87 data dan tiga transformasi yaitu *RandXReflection*, *RandXTranslation*, dan *RandYTranslation*, sehingga nantinya total data yang dihasilkan dengan proses *data augmentation* ini sebanyak $87 \times 3 = 261$ data untuk setiap iterasi *epoch*, karena terdapat 5 epoch, maka akan menghasilkan total variasi data sebanyak $261 \times 5 = 1305$ data. *Data augmentation* juga melakukan proses *resize* untuk mengubah *size* gambar menjadi 299x299 piksel, hal ini dilakukan karena *InputLayer* arsitektur *InceptionV3* membutuhkan citra dengan ukuran 299x299 piksel. Hasil preview *augmentasi* data (delapan gambar) dapat dilihat pada Gambar 3.8. dan konfigurasi *augmentasi* pada setiap data *training* dapat dilihat pada Gambar 3.9.



Gambar 3.8 Hasil *augmentasi* dengan parameter berbeda

1x1 imageDataAugmenter	
Property ▲	Value
FillValue	0
RandXReflection	1
RandYReflection	0
RandRotation	[0,0]
RandScale	[1,1]
RandXScale	[1,1]
RandYScale	[1,1]
RandXShear	[0,0]
RandYShear	[0,0]
RandXTranslation	[-30,30]
RandYTranslation	[-30,30]

Gambar 3.9 Konfigurasi *Augmentasi* pada Setiap *Data Training*

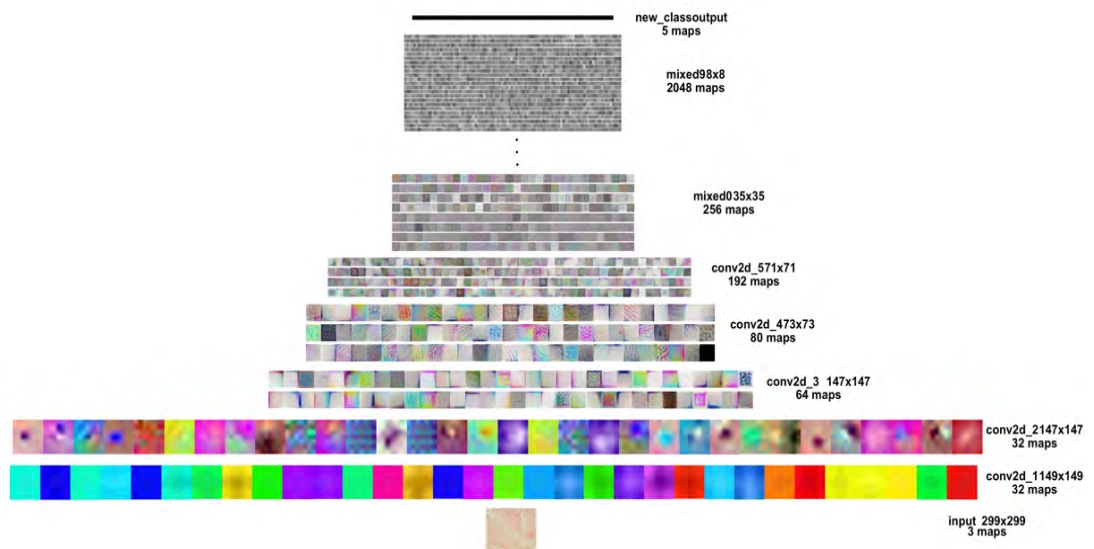
Berikut *source code* untuk proses *DataAugmentation* :

```
%data training
pixelRange = [-30 30];
imageAugmenter = imageDataAugmenter( ...
    'RandXReflection',true, ...
    'RandXTranslation',pixelRange, ...
    'RandYTranslation',pixelRange);
augimdsTrain =
augmentedImageDatastore(inputSize(1:2),imdsTrain, ...
    'DataAugmentation',imageAugmenter);
%data validasi
augimdsValidation =
augmentedImageDatastore(inputSize(1:2),imdsValidation);
%preview
%minibatch = preview(augimdsTrain);
%imshow(imtile(minibatch.input));
```

4. *Feature Extraction*

Proses ekstraksi fitur dalam arsitektur *InceptionV3* menggunakan proses *convolutional neural network* (CNN). *Feature extraction* ini terjadi pada *layer convolution2dlayer*. Dalam prosesnya, pertama-tama gambar RGB dengan resolusi 299x299 piksel digunakan sebagai citra input untuk proses *transfer learning-based model*. *Transfer learning-based model* ini terdiri dari blok konvolusi *basic*, *inception modules*, dan *softmax classifier*. Blok konvolusi *basic* ini terdiri dari 94 buah *convolution2dlayer*, mulai dari *conv2d_1* sampai *conv2d_5*. Tiap lapisan layer dalam blok konvolusi *basic* ini memiliki fungsi berbeda-beda. Blok konvolusi *conv2d_1* memiliki 32 filter dengan ukuran 3x3 yang digunakan untuk mengekstraksi *low-level features* yang menghasilkan 32 *channel feature maps* dengan ukuran 149x149. Sama halnya dengan 32,64,80 dan 192 *channel-feature maps* yang dihasilkan oleh blok konvolusi selanjutnya dengan ukuran 147x147, 147x147, 73x73 dan 71x71.

Inception modules adalah inti blok dari *transfer learning* model ini. Terdapat 11 buah *inception module*, mulai dari *mixed0* sampai *mixed10*. 256 *channel feature* dengan resolusi 35x35 dihasilkan oleh *inception module* yang berada pada *layer mixed0*. Selanjutnya proses konvolusi akan makin berkembang seiring dengan bertambahnya lapisan *layer*, sehingga akan mengekstraksi fitur yang semakin *abstract*, *layer* konvolusi ini akan menghasilkan 768 dan 2048 *channel-feature maps* dengan ukuran 17x17 dan 8x8. Terakhir, semua hasil ekstraksi fitur digabungkan menjadi sebuah vektor satu dimensi dengan output lima *neuron* sesuai dengan banyaknya kelas data yaitu lima kelas bakteri. *Neuron* dengan kemungkinan tertinggi akan menjadi *class label* dari citra bakteri yang digunakan sebagai data uji. Visualiasi proses konvolusi dan *feature mapping* dapat dilihat pada Gambar 3.10. Selain itu, informasi lengkap mengenai parameter proses konvolusi pada Gambar 3.10 dapat dilihat pada Tabel 3.1.



Gambar 3.10 Visualisasi Proses Konvolusi dan *Feature mapping*

Tabel 3.1 Parameter Konvolusi *Transfer Learning-based model InceptionV3*

Layer	Maps & Size	Filter Size
Input	3 & 299 x 299	-
Conv2d_1	32 & 149 x 149	3 x 3
Conv2d_2	32 & 147 x 147	3 x 3
Conv2d_3	64 & 147 x 147	3 x 3
Conv2d_4	80 & 73 x 73	1 x 1
Conv2d_5	192 & 71 x 71	3 x 3
Mixed0	256 & 35 x 35	1 x 1, 3 x 3
Mixed1	288 & 35 x 35	1 x 1, 3 x 3, 5 x 5
Mixed2	288 & 35 x 35	1 x 1, 3 x 3
Mixed3	768 & 17 x 17	1 x 1, 3 x 3
Mixed4	768 & 17 x 17	1 x 1, 3 x 3, 1 x 7, 7 x 1
Mixed5	768 & 17 x 17	1 x 1, 3 x 3, 1 x 7, 7 x 1
Mixed6	768 & 17 x 17	1 x 1, 3 x 3, 1 x 7, 7 x 1
Mixed7	768 & 17 x 17	1 x 1, 3 x 3, 1 x 7, 7 x 1
Mixed8	1280 & 8 x 8	1 x 1, 3 x 3, 1 x 7, 7 x 1
Mixed9	2048 & 8 x 8	1 x 1, 3 x 3, 1 x 3, 3 x 1
Mixed10	2048 & 8 x 8	1 x 1, 3 x 3, 1 x 3, 3 x 1
New_classoutput	5 x 5 x 1	-

Berikut *source code* untuk visualisasi proses konvolusi :

```
layer = 2;
name = net.Layers(layer).Name
channels = 1:32;
I = deepDreamImage(net,name,channels, ...
    'Verbose',false, ...
    'PyramidLevels',1);
figure
I = imtile(I, 'ThumbnailSize', [128 128]);
imshow(I)
title(['Layer ',name,' Features'],'Interpreter','none')
```

5. Training

Proses *training* dilakukan menggunakan metode *transfer learning* dengan konfigurasi *minibatchsize* 5, *epoch* 5, dan *learning rate* 0,0001. Banyak data *training* dan *validation* yang digunakan sesuai dengan pembagian *dataset* melalui proses *k-fold cross validation* dengan $k=5$, sehingga akan terjadi proses perulangan training sebanyak lima kali.

Berikut *source code* untuk konfigurasi *training*

```
options = trainingOptions('sgdm', ...
    'MiniBatchSize',5, ...
    'MaxEpochs',5,...
    'InitialLearnRate',1e-4, ...
    'Shuffle','every-epoch', ...
    'ValidationData',augimdsValidation, ...
    'ValidationFrequency',3, ...
    'Verbose',false, ...
    'executionEnvironment','cpu',...
    'Plots','training-progress');
```

Adapun penjelasan dari parameter-parameter yang digunakan dalam konfigurasi proses pelatihan:

1. *MiniBatchSize*

Parameter ini merupakan ukuran dari *mini-batch* yang digunakan pada setiap iterasi proses *training*. *Mini-batch* adalah bagian dari proses *training* yang digunakan untuk mengevaluasi *gradient* dari *loss function* dan melakukan *update* pada *weights*.

2. *MaxEpoch*

Parameter ini merupakan jumlah maksimal dari *epoch* yang digunakan dalam proses *training*. Iterasi merupakan Langkah yang diambil oleh *gradient descent algorithm* untuk meminimalisir *loss function* dengan *batch*. *Epoch* merupakan jalur penuh dari algoritma *training* di seluruh set *training*.

3. *InitialLearnRate*

Parameter ini menunjukkan *initiallearnrate* yang digunakan dalam proses *training*. Terdapat tiga macam *solver* yang dapat digunakan, yaitu *sgdm*, *rmsprop* dan *adam*, *solver* yang digunakan pada proses *training* ini adalah *solver sgdm*. Ketika *learningrate* terlalu rendah, proses *training* akan semakin lama. Sebaliknya, ketika *learningrate* terlalu tinggi, proses *training* akan semakin cepat, tetapi hasil *training* mungkin memberikan hasil yang kurang optimal atau menyimpang.

4. *ValidationFrequency*

Parameter ini menunjukkan jumlah iterasi yang dilakukan untuk proses validasi *network* dalam setiap *epoch*.

5. Model

Output dari proses *training* merupakan sebuah model yang nantinya akan digunakan untuk proses *validation* dengan menggunakan data *validation* yang telah dibagi melalui proses *k-fold cross validation*.

Proses training dipantau dengan tampilan *graph* yang menampilkan informasi seperti *training cycle*, *validation*, dan informasi lain seperti *hardware resource*, dll. *Graph* dapat dilihat pada Gambar 3.11.



Gambar 3.11 *Graph* Proses Training

Berikut penjelasan informasi yang ditampilkan oleh *graph* proses *training* pada Gambar 3.11:

1. *Validation accuracy*

Akurasi klasifikasi dari seluruh dataset validation.

2. *Training time*

Waktu dimulainya proses *training* dan total waktu yang berlalu mulai dari awal dijalankan proses *training* hingga proses *training* selesai. Waktu training dipengaruhi dengan kemampuan hardware yang digunakan pada proses training sistem, *resource* CPU/GPU yang digunakan ketika training, arsitektur, konfigurasi *training* dan jumlah dataset.

3. *Training cycle*

Merupakan informasi proses perulangan dari *epoch*, *iteration*, *iteration per epoch* dan *maximum iterations*.

6. RNG (*Random Number Generator*)

Umumnya dalam sistem *deep learning* yang menggunakan matlab, hasil akurasi sistem setiap kali dijalankan dengan memuat *network* yang telah jadi, akan memberikan hasil yang bervariasi karena nilai *weights* dan *bias* yang ditentukan pada awal proses eksekusi *network* ditetapkan secara *random*. *RNG default* ini digunakan agar nantinya hasil akurasi sistem setiap kali dieksekusi ulang akan menjadi konstan atau tetap.

RNG (*Random Number Generator*) digunakan dengan menambahkan parameter :

rng default

3.5.2. Proses Validation

1. *Input Citra*

Data yang digunakan diambil dari sumber yang sama dari data *training*, yaitu melalui *dataset* (<http://misztal.edu.pl/software/databases/dibas/>) yang kemudian akan dibagi pada proses *preprocessing*.

2. *Preprocessing*

Tahapan *preprocessing* sampai *feature extraction* sama dengan proses yang dilalui oleh data *training*.

3. *Classification*

Klasifikasi dilakukan dengan parameter :

```
[YPred,scores] = classify(net,augimdsValidation);
```

Fungsi *classify* akan menghasilkan matrix *scores* dengan ukuran 21x5 yang berisi akurasi dari lima kelas *dataset*. Kelas dengan akurasi tertinggi akan digunakan sebagai *label* data dari data validation yang telah diklasifikasi kemudian disimpan pada matriks *ypred* dengan ukuran 21x1.

Ketika dieksekusi, fungsi *classify* akan mengklasifikasi data *validation* yang berada pada *cell augimdsvalidation* menggunakan *network net*, *cell* ini berisi data *validation* yang berjumlah 21 gambar. *Label* merupakan matriks berukuran N-dari-1 dimana N merupakan jumlah data *validation*, dan *scores* merupakan matriks berukuran N-dari-C, dimana N merupakan jumlah data *validation* dan C merupakan jumlah kelas dari datasetnya.

Proses perhitungan skor probabilitas pada multiclass dengan menggunakan fungsi *classify* adalah sebagai berikut.

$$P(c_r | x, \theta) = \frac{P(x, \theta | c_r) P(c_r)}{\sum_{j=1}^k P(x, \theta | c_j) P(c_j)} = \frac{\exp(a_r(x, \theta))}{\sum_{j=1}^k \exp(a_j(x, \theta))} \quad (3.1)$$

Dimana $0 \leq P(c_r | x, \theta) \leq 1$ dan $\sum_{j=1}^k P(c_j | x, \theta) = 1$

Selain itu, a_r merupakan probabilitas bersyarat dari sampel yang oleh kelas r dan $P(c_r)$ adalah probabilitas kelas sebelumnya. Proses perhitungan skor probabilitas berada pada rentang $0 - 1$, dan kelas dengan probabilitas tertinggi akan sebagai *label* dari data *validation*.

Untuk perhitungan waktu eksekusi dari proses validasi, digunakan parameter “tic” yang akan memulai *stopwatch timer*, dan parameter “toc” untuk membaca waktu yang telah berlalu dari *stopwatch* sehingga dalam penggunaannya, parameter ini akan ditempatkan seperti berikut:

```
tic
% kode program dari awal hingga proses klasifikasi
toc
```

Kemudian akan memberikan output seperti berikut:

```
Elapsed time is 46.158938 seconds.
```

Berikut potongan *source code* untuk fungsi *classify*:

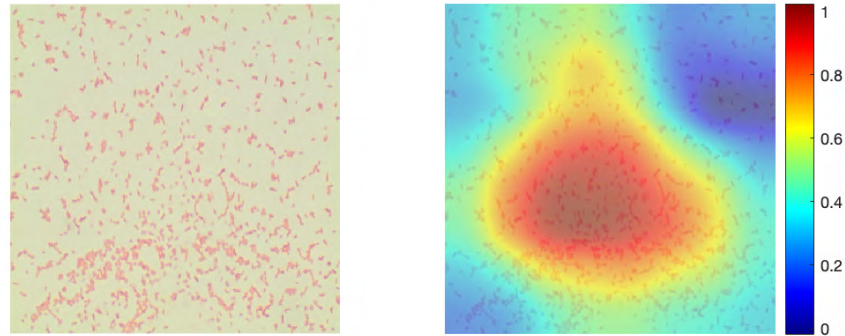
```
try
    % 'calculatePredict' will return a cell array
    as it supports multiple
        % output networks. But 'classify' only
    supports single output networks, so
        % we know the output will be a 1-by-1 cell
    array.
    scores = this.calculatePredict( ...
        X, options, executionEnvironment,
    acceleration );
    scores = scores{1};
catch ex
    throw(ex);
end

if ~this.NetworkInfo.HasSequenceOutput
    labels = iUndummify( scores,
        layers{this.PrivateNetwork.OutputLayerIndices(1)}.
    Categories );
else
    labels = iUndummifySequence( scores,
        layers{this.PrivateNetwork.OutputLayerIndices(1)}.
    Categories );
end
end
```

4. Visualization

Untuk melihat proses klasifikasi, digunakan metode *gradient-weighted class activation mapping (Grad-CAM)*. *Grad-CAM* menggunakan *gradient* dari *score* klasifikasi dengan fitur konvolusional yang telah ditentukan oleh *network model* untuk mengetahui bagian pada gambar yang berperan penting dalam proses klasifikasi. Hasil visualisasi dapat dilihat pada Gambar 3.12.

Escherichia



Gambar 3.12 Visualiasi *Grad-CAM*

Heatmap Color Scale dari visualisasi *Grad-CAM* diperlihatkan dengan skala 0 sampai 1, berikut penjelasannya:

0 - 0.2, bagian visualisasi kecil kemungkinan berperan penting dalam proses klasifikasi.

0.2 - 0.4, bagian visualisasi sedikit berperan penting dalam proses klasifikasi.

0.4 - 0.6, bagian visualisasi cukup berperan penting dalam proses klasifikasi.

0.6 - 0.8, bagian visualiasi berperan penting dalam proses klasifikasi.

0.8 - 1, bagian visualisi sangat berperan penting dalam proses klasifikasi.

Berikut *source code* untuk visualisasi *Grad-CAM* :

```
lgraph = layerGraph(net);
lgraph = removeLayers(lgraph,
lgraph.Layers(end).Name);
dlnet = dlnetwork(lgraph);
softmaxName = 'predictions_softmax';
convLayerName = 'conv2d_94';

for i = 1:numel(imdsValidation.Files)
    imageNumber = i;

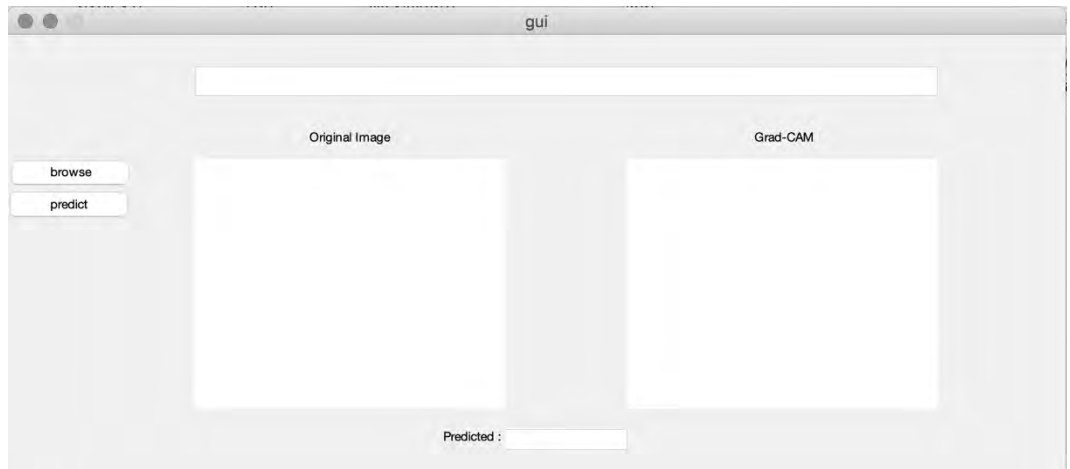
    observation =
augimdsValidation.readByIndex(idx(imageNumber));
    img = observation.input{1};
    label = YPred(idx(imageNumber));
    gradcamMap = computeGradCAM(dlnet, img,
softmaxName, convLayerName, label);

    figure
    alpha = 0.5;
    plotGradCAM(img, gradcamMap, alpha);
    sgttitle(string(label));
end
```

3.5.3. Proses Testing

1. GUI (*Graphical User Interface*)

GUI dalam sistem ini berperan untuk memudahkan pengguna dalam melakukan klasifikasi bakteri, pengguna cukup *menginput*, gambar bakteri yang akan diklasifikasi dengan cara mengklik tombol “*browse*”, kemudian mengklik tombol “prediksi”. Sistem akan mengklasifikasi bakteri berdasarkan *network* yang telah dilatih, dan akan memberikan output berupa hasil prediksi, persentase akurasi, dan visualisasi Grad-CAM. Tampilan dari GUI ini dapat dilihat pada Gambar 3.13.



Gambar 3.13 GUI (*Graphical User Interface*)

2. *Input Citra*

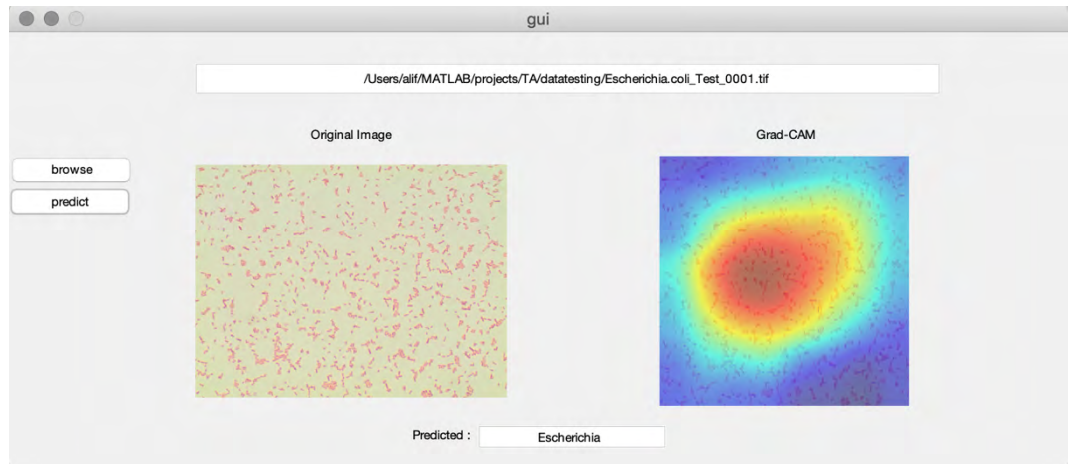
Data yang digunakan untuk proses testing berbeda dari dataset yang ada pada data *training* dan *validation*. Untuk setiap sesi proses testing, jumlah maksimal yang dapat diklasifikasi adalah satu buah gambar bakteri.

3. *Classification*

Tahapan proses *classification* sama dengan proses yang dilalui oleh data *validation*.

4. *Visualization*

Hasil klasifikasi akan menampilkan gambar original, visualisasi Grad-CAM, serta hasil prediksi dan akurasi dari *input* data testing. Visualisasi dapat dilihat pada Gambar 3.14.



Gambar 3.14 Visualisasi Data Testing

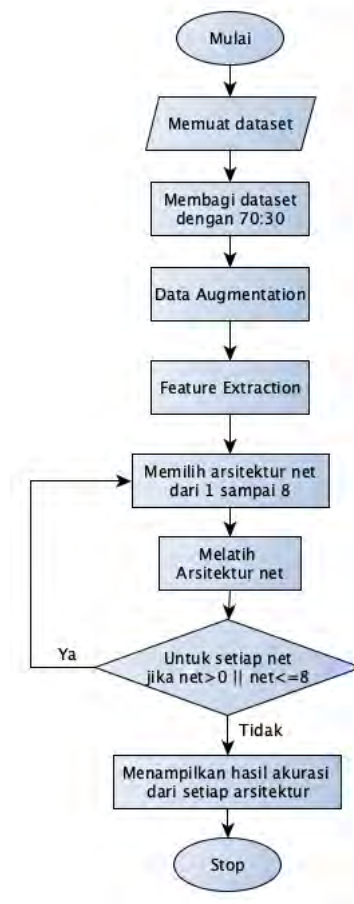
Berikut potongan *source code* untuk visualisasi data testing :

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a future
version of MATLAB
% handles      structure with handles and user data
(see GUIDATA)
handles = guidata(hObject);
net = handles.net;
I = handles.I;
Image= imresize(I,[299,299],'nearest');
[Pred,scores] = classify(net,Image);
scores = max(double(scores*100));
set(handles.edit2,'string',join([string(Pred),' '
,scores ,'%']));

%gradcam
lgraph = layerGraph(net);
lgraph = removeLayers(lgraph,
lgraph.Layers(end).Name);
dlnet = dlnetwork(lgraph);
softmaxName = 'predictions_softmax';
featureLayerName = 'conv2d_94';
```

3.6. Perbandingan Sistem InceptionV3 dengan Tujuh Arsitektur Lain

Untuk melihat perbandingan dari performa arsitektur InceptionV3 dengan arsitektur yang lain, maka akan dilakukan percobaan dengan membuat program berdasarkan *flowchart* pada Gambar 3.15.



Gambar 3.15 *Flowchart* Program Perbandingan InceptionV3 dengan tujuh arsitektur lain

3.7. Analisis Kerja Sistem

Untuk melihat kinerja dari sistem klasifikasi bakteri dapat dilakukan dengan menghitung performa prediksi yang dilakukan sistem terhadap data citra inputan pada proses *training* dengan menggunakan bantuan *confusion matrix*.

Confusion Matrix terdiri atas empat bagian yaitu:

- True Positive

Merupakan jumlah prediksi benar dari data positif.

- False Positive

Merupakan jumlah prediksi salah dari data positif.

- True Negative

Merupakan jumlah prediksi benar dari data negatif.

- False Negative

Merupakan jumlah prediksi salah dari data negatif.

Untuk menghitung nilai akurasi sistem, dapat dilakukan dengan persamaan:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Sistem klasifikasi bakteri dengan *deep learning* ini menggunakan metode *transfer learning-based model* dengan arsitektur *inceptionv3*. Data yang digunakan berjumlah 108 citra yang terdiri dari lima kelas bakteri yaitu *Acinetobacter baumannii*, *Escherichia coli*, *Neisseria gonorrhoeae*, *Propionibacterium acnes* dan *Veionella*. Data ini kemudian dibagi menjadi data *training* dan *validation* menggunakan metode *k-fold cross validation* dengan jumlah *fold* $k=5$ yang menghasilkan 87 citra untuk data *training* dan 21 citra untuk data *validation*. Sebelum data *training*, dilakukan proses *data augmentation* dengan transformasi *RandXReflection*, *RandXTranslation*, dan *RandYTranslation* sehingga proses *training* akan memberikan lebih banyak variasi dataset kedalam *network*.

Pada penelitian ini dilakukan empat percobaan dimulai dengan percobaan pertama menggunakan arsitektur *inceptionv3* dengan *k-fold cross validation*, percobaan kedua yaitu menggunakan arsitektur *inceptionv3* tanpa *k-fold cross validation* lalu membagi data *training* dan *validation* dengan persentase 70:30, percobaan ketiga yaitu menggunakan arsitektur *inceptionv3* dengan konfigurasi *rng default*, percobaan keempat yaitu menggunakan arsitektur *inceptionv3* dengan *k-fold cross validation* tanpa proses *augmentation*, dan terakhir dilakukan perbandingan performa arsitektur *inceptionv3* dengan tujuh arsitektur lain yaitu *googlenet*, *vgg16*, *vgg19*, *resnet101*, *inceptionresnetv2*, *squeezenet* dan *xception*.

Konfigurasi *training* dalam setiap percobaan menggunakan parameter yang sama, yaitu *minibatchsize*, *maxepochs*, *initiallearnrate*, dan *validationfrequency*. Nilai yang digunakan untuk masing-masing parameter ini adalah *minibatchsize* 5, *maxepoch* 5, *initiallearnrate* 0,0001, dan *validationfrequency* 3. Setiap percobaan di eksekusi sepuluh kali untuk melihat rata-rata performanya, dan diuji dengan data testing.

4.1.1. Hasil Percobaan Pertama

Pada percobaan pertama, sistem yang dibuat menggunakan arsitektur *inceptionv3* dengan *k-fold cross validation*. Adapun Hasil prediksi terhadap lima jenis bakteri dari percobaan sistem *InceptionV3* dengan *k-fold cross validation* dapat dilihat pada Tabel 4.1.

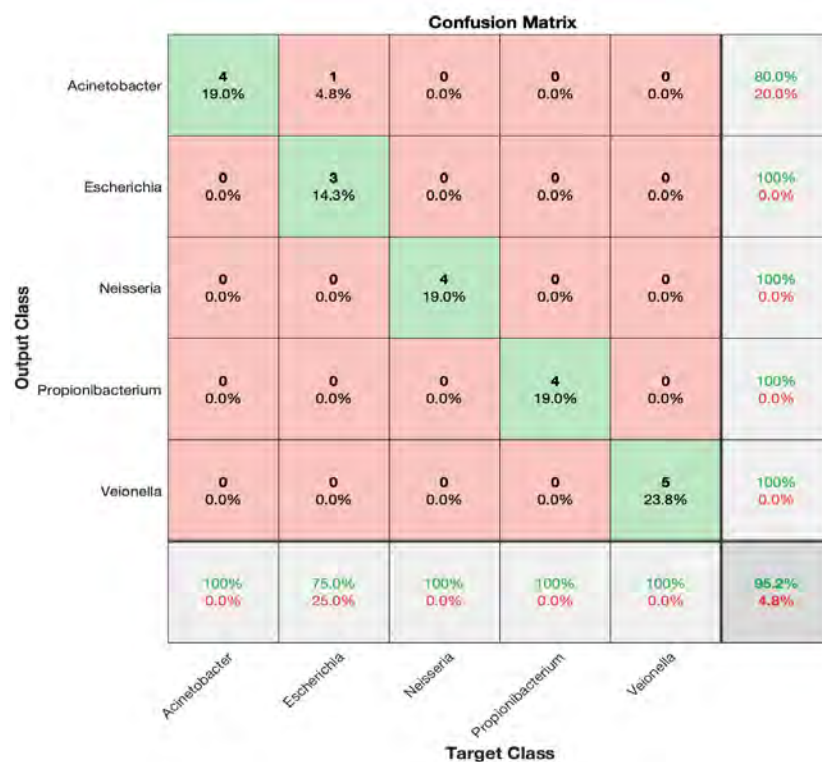
Tabel 4.1 Hasil Prediksi Sistem *InceptionV3* dengan *k-fold cross validation*

Data	Acinetobacter	Escherichia	Neisseria	Propionibacterium	Veionella
1	0.75095880	0.080308117	0.12230612	0.018657329	0.027769702
2	0.84342718	0.070955619	0.059331294	0.0033539920	0.022931822
3	0.69347358	0.15840755	0.083476648	0.014401862	0.050240349
4	0.46718633	0.34495175	0.10172542	0.021079296	0.065057181
5	0.12573697	0.71777040	0.068349615	0.015915072	0.072227955
6	0.083378099	0.81271881	0.053948741	0.018019291	0.031935081
7	0.099736191	0.82626390	0.020875424	0.019054854	0.034069624
8	0.41302285	0.28462869	0.10016134	0.026424734	0.17576240
9	0.13306879	0.027085969	0.79402280	0.0091096647	0.036712788
10	0.23666957	0.049537383	0.61832738	0.026968848	0.068496734
11	0.072292008	0.13126534	0.74097288	0.019277481	0.036192223
12	0.27976984	0.025963923	0.62077093	0.015348235	0.058147125
13	0.025836732	0.026531056	0.062287338	0.68208420	0.20326068
14	0.11945159	0.20255961	0.11562056	0.47290409	0.089464158
15	0.019457703	0.052123804	0.016546162	0.88146019	0.030412188
16	0.083030291	0.11498720	0.16297045	0.34621769	0.29279438
17	0.074518695	0.050949208	0.023857212	0.035250060	0.81542486

Data	Acinetobacter	Escherichia	Neisseria	Propionibacterium	Veionella
18	0.089701138	0.036186866	0.059155520	0.061379705	0.75357676
19	0.094259702	0.045518633	0.038290400	0.054680582	0.76725066
20	0.030813972	0.060877562	0.039209273	0.015125679	0.85397357
21	0.11716594	0.077739432	0.047611341	0.10984911	0.64763421

Tabel diatas menunjukkan hasil prediksi data *validation* terhadap lima jenis bakteri. Jenis bakteri dengan akurasi tertinggi akan menjadi *label* dari data *validation* tersebut yang ditandai dengan tulisan berwarna merah pada tabel.

Performa dari sistem dilihat dengan menggunakan *confusion matriks*. *Confusion matriks* untuk arsitektur *inceptionv3* dan *k-fold cross validation* dapat dilihat pada Gambar 4.1



Gambar 4.1 *Confusion Matriks* Arsitektur *Inceptionv3* dengan *K-fold cross validation*

Berdasarkan Gambar 4.1 diatas, dapat dilihat akurasi untuk arsitektur *inceptionv3* dengan *k-fold cross validation* sebesar 95.2%.

Akurasi ini dapat dihitung dengan persamaan :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4.1)$$

Sehingga :

$$Accuracy = \frac{(4+3+4+4+5)+(0)}{(4+3+4+4+5)+(0)+(1)+(0)} = \frac{20}{21} = 0.952$$

Hasil eksekusi percobaan sebanyak sepuluh kali dapat dilihat pada Tabel 4.2.

Tabel 4.2 Validasi *InceptionV3* dengan *K-fold cross validation*

Percobaan	Waktu Eksekusi	Akurasi (%)
1	44 detik	95.45%
2	46 detik	90.48%
3	44 detik	95.45%
4	44 detik	95.45%
5	42 detik	95.29%
6	49 detik	95.37%
7	47 detik	92.51%
8	43 detik	94.50%
9	42 detik	95.41%
10	44 detik	94.42%
Rata – rata Waktu Eksekusi		44 detik
Rata - rata Akurasi		94.43%

Dari percobaan yang dilakukan pada Tabel 4.2 dapat dilihat, akurasi tiap percobaan berbeda-beda, hal ini terjadi karena setiap kali dijalankan, nilai awal *weight* dan *bias* yang diberikan ke model berbeda-beda. Nilai *weight* dan *bias* ini mempengaruhi hasil akurasi dari setiap percobaan tersebut, Adapun untuk percobaan dengan nilai akurasi tertinggi yaitu 95.45%, model yang terbentuk memberi nilai awal *weight* 3x3x32 dan *bias* 1x1x32 pada konvolusi “conv2d_1” seperti yang ditunjukkan pada Gambar 4.2

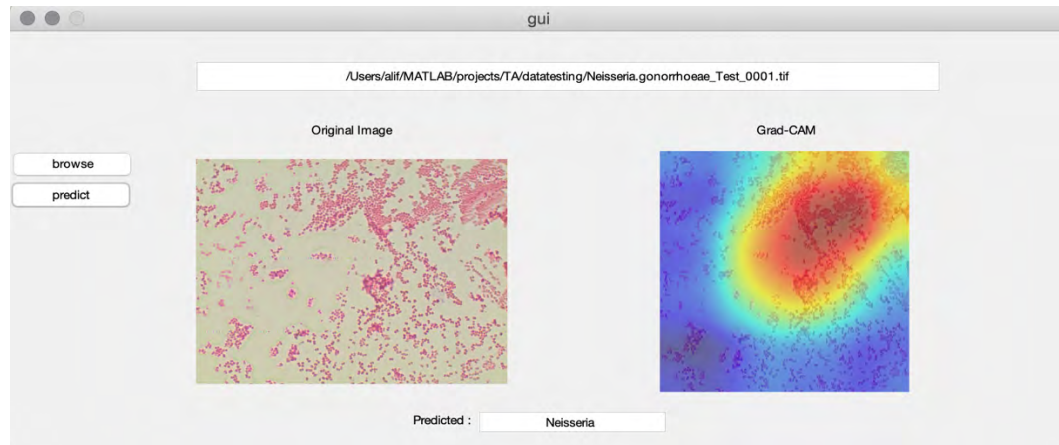
net
Analysis date: 14-Feb-2021 16:52:48

315 layers 0 warnings 0 errors

ANALYSIS RESULT				
	Name	Type	Activations	Learnables
1	input_1 299×299×3 images with 'rescale-sy...	Image Input	299×299×3	-
2	conv2d_1 32 3×3×3 convolutions with stride [...]	Convolution	149×149×32	Weights 3×3×3×32 Bias 1×1×32
3	batch_normalization_1 Batch normalization with 32 channels	Batch Normalization	149×149×32	Offset 1×1×32 Scale 1×1×32
4	activation_1_relu ReLU	ReLU	149×149×32	-
5	conv2d_2 32 3×3×32 convolutions with stride [...]	Convolution	147×147×32	Weights 3×3×32×32 Bias 1×1×32
6	batch_normalization_2 Batch normalization with 32 channels	Batch Normalization	147×147×32	Offset 1×1×32 Scale 1×1×32
7	activation_2_relu ReLU	ReLU	147×147×32	-
8	conv2d_3 64 3×3×32 convolutions with stride [...]	Convolution	147×147×64	Weights 3×3×32×64 Bias 1×1×64
9	batch_normalization_3 Batch normalization with 64 channels	Batch Normalization	147×147×64	Offset 1×1×64 Scale 1×1×64
10	activation_3_relu ReLU	ReLU	147×147×64	-
11	max_pooling2d_1 3×3 max pooling with stride [2,2] a...	Max Pooling	73×73×64	-
12	conv2d_4 80 1×1×64 convolutions with stride [...]	Convolution	73×73×80	Weights 1×1×64×80 Bias 1×1×80
13	batch_normalization_4 Batch normalization with 80 channels	Batch Normalization	73×73×80	Offset 1×1×80 Scale 1×1×80
14	activation_4_relu ReLU	ReLU	73×73×80	-
15	conv2d_5	Convolution	71×71×192	Weights 3×3×80×192

Gambar 4.2 Nilai Awal *Weight* dan *Bias* Model Terbaik

Selanjutnya dilakukan percobaan klasifikasi data *testing* pada sistem GUI menggunakan *network* yang telah dibuat pada percobaan ini. Hasilnya dapat dilihat pada Gambar 4.3.



Gambar 4.3 Klasifikasi Data Testing Percobaan Pertama

4.1.2. Hasil Percobaan Kedua

Pada percobaan kedua, sistem yang dibuat menggunakan arsitektur *inceptionv3* tanpa *k-fold cross validation*. Data *training* dan *testing* dibagi dengan persentase 70:30.

Performa dari sistem dilihat dengan menggunakan *confusion matriks*. *Confusion matriks* untuk arsitektur *inceptionv3* tanpa *k-fold cross validation* dapat dilihat pada Gambar 4.4.

Confusion Matrix							
Output Class	Acinetobacter	6 18.2%	2 6.1%	0 0.0%	0 0.0%	0 0.0%	75.0% 25.0%
	Escherichia	0 0.0%	4 12.1%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	Neisseria	0 0.0%	0 0.0%	7 21.2%	0 0.0%	0 0.0%	100% 0.0%
	Propionibacterium	0 0.0%	0 0.0%	0 0.0%	7 21.2%	0 0.0%	100% 0.0%
	Veilonella	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 21.2%	100% 0.0%
		100% 0.0%	66.7% 33.3%	100% 0.0%	100% 0.0%	100% 0.0%	93.8% 6.1%
		Acinetobacter	Escherichia	Neisseria	Propionibacterium	Veilonella	
		Target Class					

Gambar 4.4 *Confusion Matriks* Arsitektur *Inceptionv3* tanpa *K-fold cross validation*

Berdasarkan Gambar 4.4 diatas, dapat dilihat akurasi untuk arsitektur *inceptionv3* tanpa *k-fold cross validation* sebesar 93.9%.

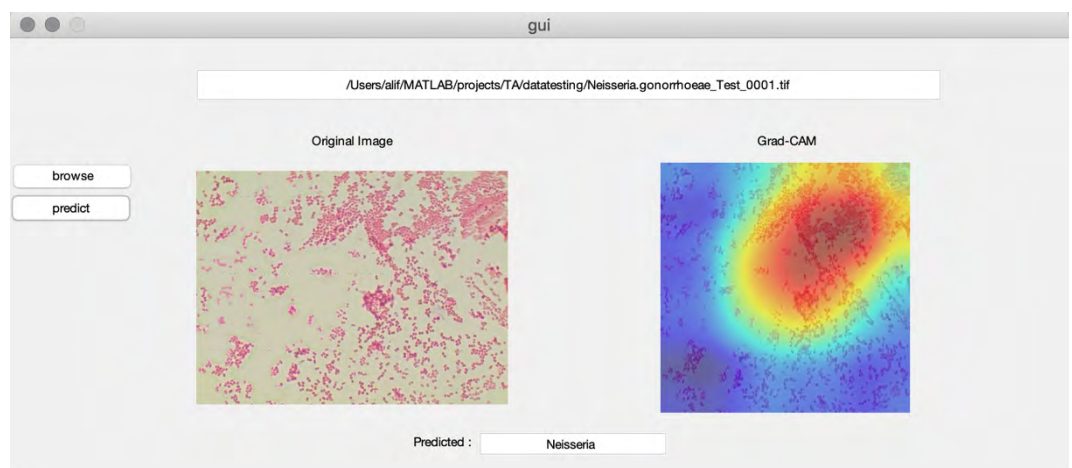
Hasil eksekusi percobaan sebanyak sepuluh kali dapat dilihat pada Tabel 4.3.

Tabel 4.3 Validasi *InceptionV3* tanpa *K-fold cross validation*

Percobaan	Waktu Eksekusi	Akurasi (%)
1	14 detik	90.91%
2	14 detik	93.94%
3	14 detik	93.94%
4	14 detik	84.85%

Percobaan	Waktu Eksekusi	Akurasi (%)
5	15 detik	81.82%
6	13 detik	93.94%
7	13 detik	87.88%
8	17 detik	84.85%
9	14 detik	93.94%
10	13 detik	93.94%
Rata – rata Waktu Eksekusi		14 detik
Rata - rata Akurasi		90.00%

Selanjutnya dilakukan percobaan klasifikasi data *testing* pada sistem GUI menggunakan *network* yang telah dibuat pada percobaan ini. Hasilnya dapat dilihat pada Gambar 4.5.

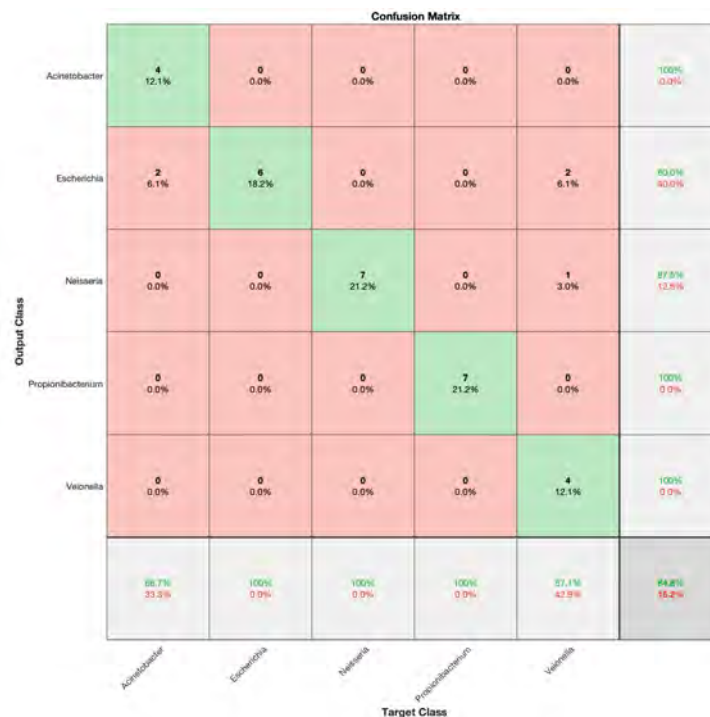


Gambar 4.5 Klasifikasi Data Testing Percobaan Kedua

4.1.3. Hasil Percobaan Ketiga

Pada percobaan ketiga, sistem yang dibuat menggunakan arsitektur *inceptionv3* dengan konfigurasi *RNG default*.

Performa dari sistem dengan arsitektur *inceptionv3* dengan konfigurasi *RNG default* dapat dilihat pada *confusion matriks* yang ada pada Gambar 4.6.



Gambar 4.6 Confusion Matriks Arsitektur *Inceptionv3* dengan konfigurasi *RNG default*

Berdasarkan Gambar 4.6 diatas, dapat dilihat akurasi untuk arsitektur *inceptionv3* dengan konfigurasi *RNG default* sebesar 84.8%.

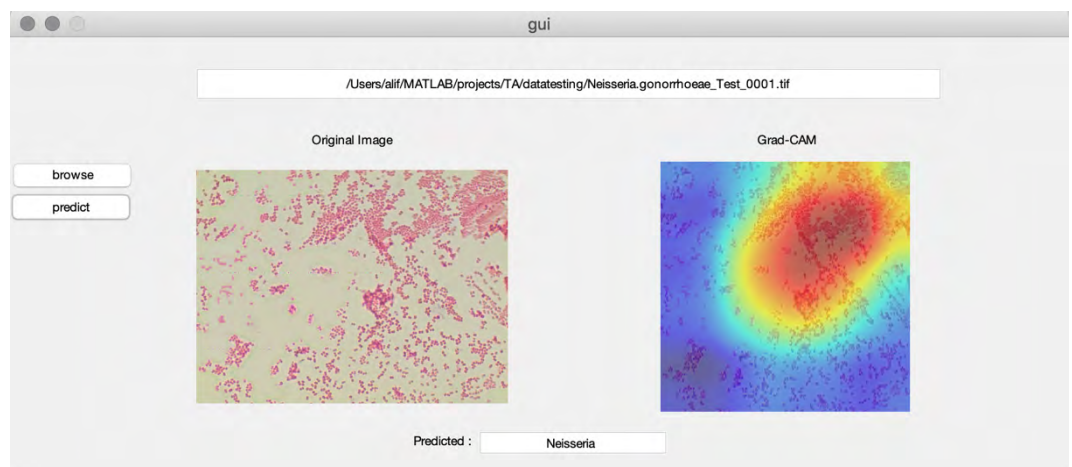
Hasil eksekusi percobaan sebanyak sepuluh kali dapat dilihat pada Tabel 4.4.

Tabel 4.4 Validasi *InceptionV3* dengan konfigurasi *RNG Default*

Percobaan	Waktu Eksekusi	Akurasi (%)
1	18 detik	84.85%

Percobaan	Waktu Eksekusi	Akurasi (%)
2	17 detik	84.85%
3	18 detik	84.85%
4	21 detik	84.85%
5	20 detik	84.85%
6	21 detik	84.85%
7	22 detik	84.85%
8	23 detik	84.85%
9	23 detik	84.85%
10	20 detik	84.85%
Rata – rata Waktu Eksekusi		20 detik
Rata - rata Akurasi		84.85%

Selanjutnya dilakukan percobaan klasifikasi data *testing* pada sistem GUI menggunakan *network* yang telah dibuat pada percobaan ini. Hasilnya dapat dilihat pada Gambar 4.7.



Gambar 4.7 Klasifikasi Data Testing Percobaan Ketiga

4.1.4. Hasil Percobaan Keempat

Pada percobaan keempat, sistem yang dibuat menggunakan arsitektur *inceptionv3* dengan *k-fold cross validation* tanpa proses *augmentation*.

Performa dari sistem dilihat dengan menggunakan *confusion matriks*. *Confusion matriks* untuk arsitektur *inceptionv3* dengan *k-fold cross validation* tanpa proses *augmentation* dapat dilihat pada Gambar 4.8.

		Confusion Matrix					
Output Class	Acinetobacter	4 19.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	Escherichia	0 0.0%	3 14.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	Neisseria	0 0.0%	1 4.8%	4 19.0%	0 0.0%	0 0.0%	80.0% 20.0%
	Propionibacterium	0 0.0%	0 0.0%	0 0.0%	4 19.0%	0 0.0%	100% 0.0%
	Veionella	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 23.8%	100% 0.0%
	100% 0.0%	75.0% 25.0%	100% 0.0%	100% 0.0%	100% 0.0%	95.2% 4.8%	
		Target Class					
		Acinetobacter	Escherichia	Neisseria	Propionibacterium	Veionella	

Gambar 4.8 Confusion Matriks Arsitektur *Inceptionv3* dengan *k-fold cross validation* tanpa *augmentation*.

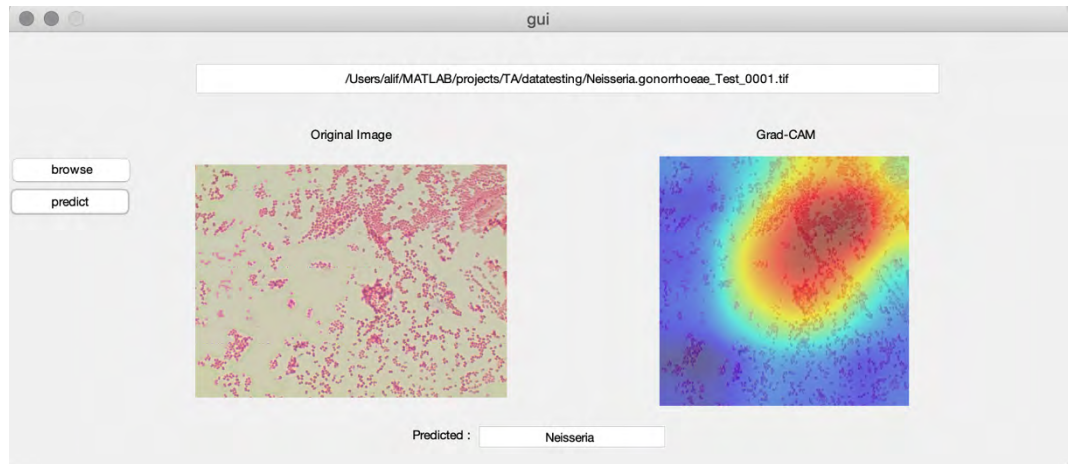
Berdasarkan Gambar 4.8 diatas, dapat dilihat akurasi untuk arsitektur *inceptionv3* dengan dengan *k-fold cross validation* tanpa *augmentation* sebesar 95.2%.

Hasil eksekusi percobaan sebanyak sepuluh kali dapat dilihat pada Tabel 4.5.

Tabel 4.5 Validasi *InceptionV3* tanpa *augmentation*

Percobaan	Waktu Eksekusi	Akurasi (%)
1	47 detik	72.73%
2	50 detik	85.71%
3	48 detik	95.45%
4	50 detik	90.91%
5	42 detik	95.24%
6	52 detik	94.46%
7	47 detik	90.78%
8	47 detik	93.51%
9	42 detik	81.82%
10	40 detik	90.91%
Rata – rata Waktu Eksekusi		46 detik
Rata - rata Akurasi		89.15%

Selanjutnya dilakukan percobaan klasifikasi data *testing* pada sistem GUI menggunakan *network* yang telah dibuat pada percobaan ini. Hasilnya dapat dilihat pada Gambar 4.9.



Gambar 4.9 Klasifikasi Data Testing Percobaan Keempat

4.1.5. Percobaan Menggunakan Data *Training* Sebagai Data Validasi

Pada percobaan ini, sistem yang dibuat menggunakan data *training* sebanyak 75 data, kemudian ke 75 data tersebut digunakan pula sebagai data validasi nya. Konfigurasi *training* dalam percobaan menggunakan parameter *minibatchsize*, *maxepochs*, *initiallearnrate*, dan *validationfrequency*. Nilai yang digunakan untuk masing-masing parameter ini adalah *minibatchsize* 15, *maxepoch* 5, *initiallearnrate* 0,0001, dan *validationfrequency* 3.

Performa dari sistem dilihat dengan menggunakan *confusion matriks*. *Confusion matriks* untuk percobaan menggunakan data *training* sebagai data validasi dapat dilihat pada Gambar 4.10.

Confusion Matrix

Output Class	Acinetobacter	15 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	Escherichia	0 0.0%	15 20.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	Neisseria	0 0.0%	0 0.0%	15 20.0%	0 0.0%	0 0.0%	100% 0.0%
	Propionibacterium	0 0.0%	0 0.0%	0 0.0%	15 20.0%	0 0.0%	100% 0.0%
	Veionella	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 20.0%	100% 0.0%
		100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	Acinetobacter	Escherichia	Neisseria	Propionibacterium	Veionella		Target Class

Gambar 4.10 *Confusion Matriks* Percobaan Data *Training* Sebagai Data Validasi

Berdasarkan Gambar 4.10 diatas, dapat dilihat akurasi untuk percobaan data *training* sebagai data validasi sebesar 100%.

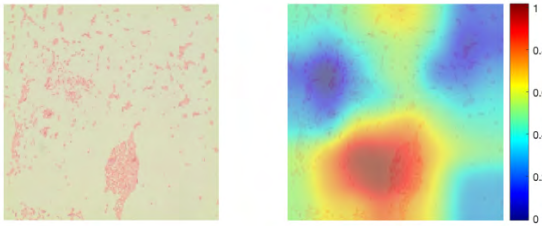
4.2. Pembahasan

Berdasarkan hasil keempat percobaan yang dilakukan, dapat dilihat dalam percobaan pertama, rata-rata akurasi lebih tinggi dibandingkan dengan percobaan kedua, hal ini terjadi karena pengaruh *k-fold cross validation* yang membuat akurasi sistem semakin naik. Dengan menggunakan *k-fold cross validation*, setiap data dalam *dataset* memiliki peluang untuk menjadi data *training* dan *validation*, sehingga didapatkan *dataset* terbaik yang menghasilkan akurasi tertinggi.

Selanjutnya pada percobaan ketiga, dibandingkan dengan percobaan kedua, sistem memberikan hasil akurasi yang tetap setiap kali *ditraining* karena dengan *rng default*, nilai *weight* dan *bias* dalam proses konvolusi menjadi konstan sehingga menyebabkan hasil akurasi ketika *ditraining* ulang akan selalu konstan. Terakhir, pada percobaan keempat, dapat dilihat jika tidak menggunakan *augmentation* pada sistem, maka akurasi sistem akan menurun karena kurang nya variasi data dalam proses training sehingga membuat *network* yang dihasilkan menjadi tidak optimal. Sistem dalam keempat percobaan ini dibuat untuk mengklasifikasi empat jenis bakteri yaitu *Acinetobacter baumanii*, *Escherichia coli*, *Neisseria gonorrhoeae*, *Propionibacterium acnes* dan *Veionella*.

Pada percobaan pertama dapat dilihat dalam *confusion matriks* terdapat satu *false positive* atau kesalahan klasifikasi, yaitu pada klasifikasi data *validation* bakteri *Escherichia coli*, yang kemudian sistem mengklasifikasikannya sebagai bakteri *acinetobacter baumanii*. Hal ini disebabkan karena kedua bakteri tersebut memiliki ciri-ciri yang mirip yaitu berwarna *gram-negative* serta memiliki bentuk yang mirip yaitu bentuk dalam kategori *bacilli*. Proses klasifikasi yang salah ini dapat dilihat dengan visualisasi *grad-cam* pada Tabel 4.6

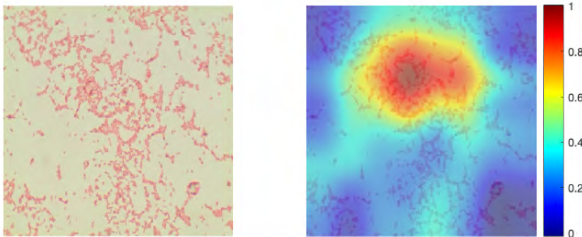
Tabel 4.6 Hasil Kesalahan Klasifikasi

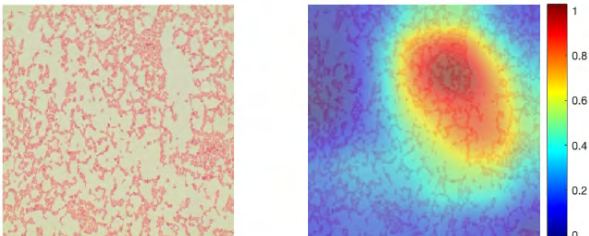
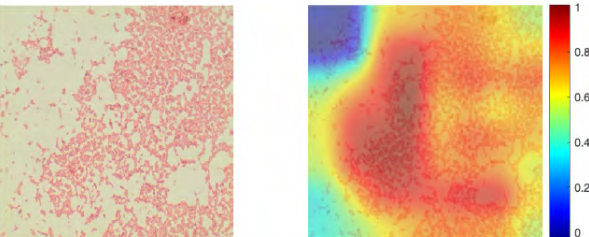
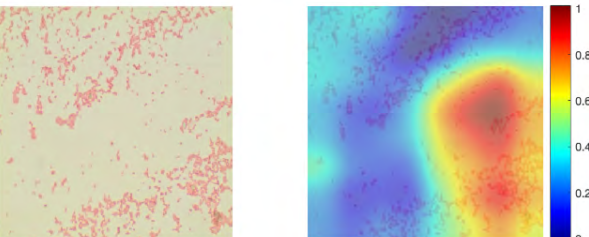
Data Validation	Visualisasi	Aktual	Prediksi
8	Acinetobacter 	Escherichia	Acinetobacter

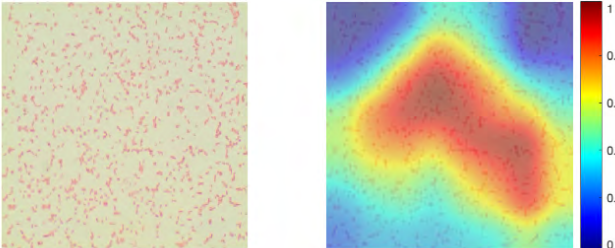
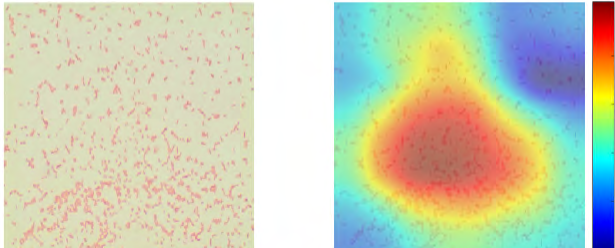
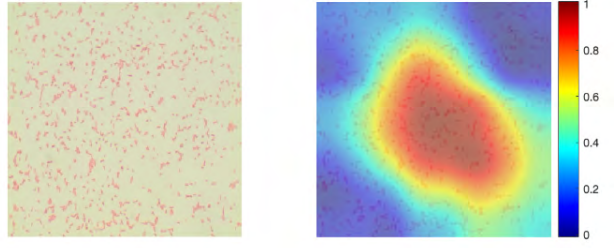
Adapun untuk visualisasi hasil klasifikasi bakteri yang benar dapat dilihat pada

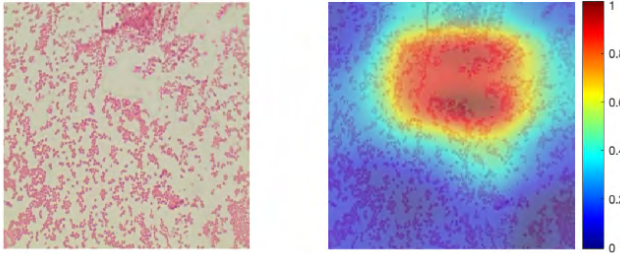
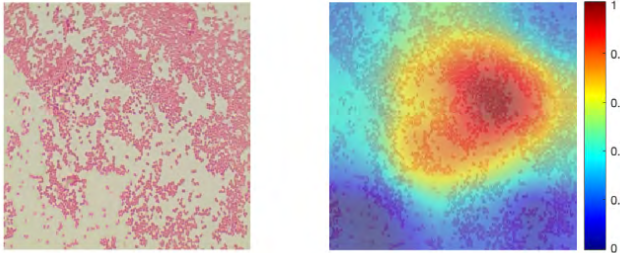
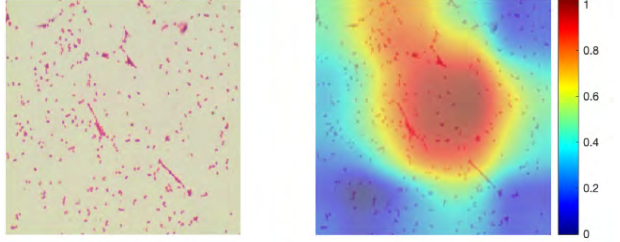
Tabel 4.7.

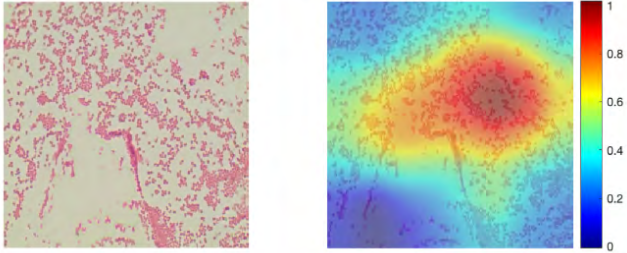
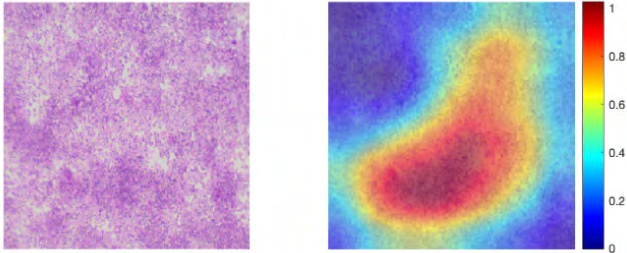
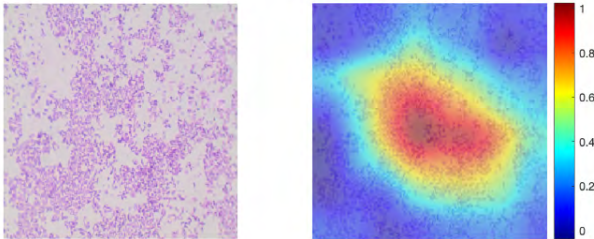
Tabel 4.7 Hasil Klasifikasi Benar

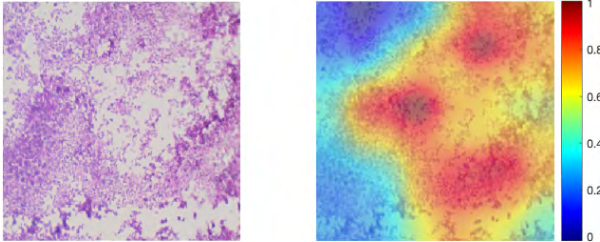
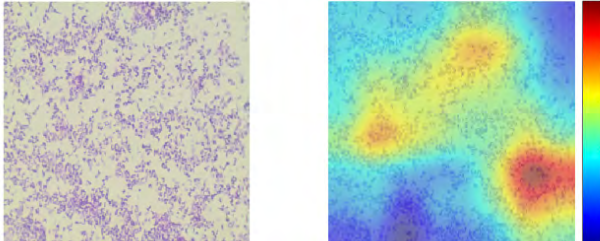
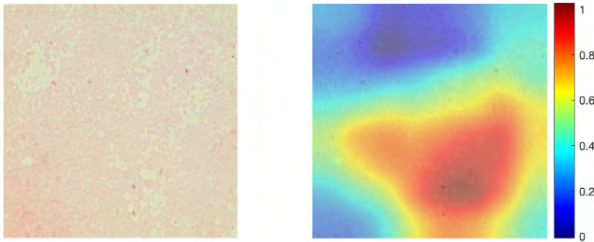
Data Validation	Visualisasi	Aktual	Prediksi
1	Acinetobacter 	Acinetobacter	Acinetobacter

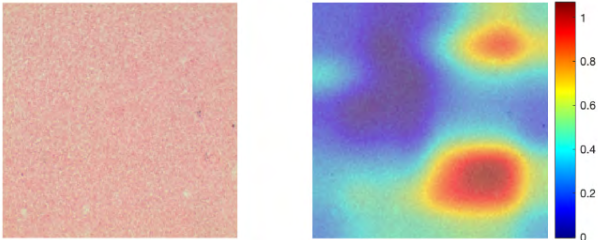
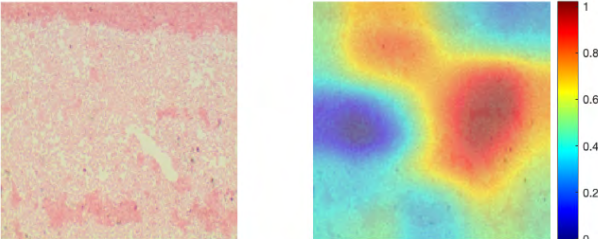
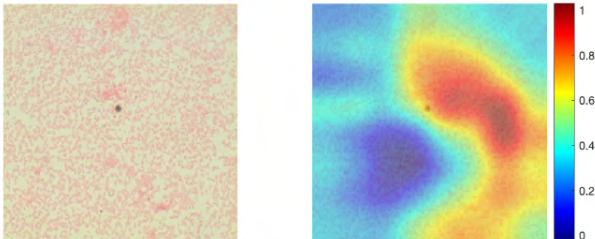
Data Validation	Visualisasi	Aktual	Prediksi
2	Acinetobacter 	Acinetobacter	Acinetobacter
3	Acinetobacter 	Acinetobacter	Acinetobacter
4	Acinetobacter 	Acinetobacter	Acinetobacter

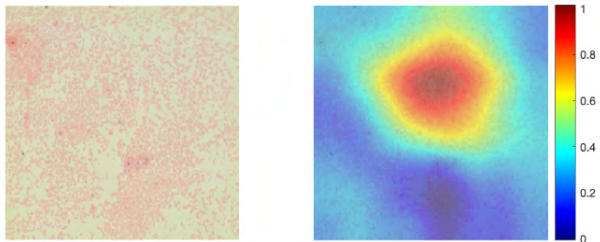
Data Validation	Visualisasi	Aktual	Prediksi
5	Escherichia 	Escherichia	Escherichia
6	Escherichia 	Escherichia	Escherichia
7	Escherichia 	Escherichia	Escherichia

Data Validation	Visualisasi	Aktual	Prediksi
9	Neisseria 	Neisseria	Neisseria
10	Neisseria 	Neisseria	Neisseria
11	Neisseria 	Neisseria	Neisseria

Data Validation	Visualisasi	Aktual	Prediksi
12	Neisseria 	Neisseria	Neisseria
13	Propionibacterium 	Propionibacte rium	Propionibacte rium
14	Propionibacterium 	Propionibacte rium	Propionibacte rium

Data Validation	Visualisasi	Aktual	Prediksi
15	Propionibacterium 	Propionibacterium	Propionibacterium
16	Propionibacterium 	Propionibacterium	Propionibacterium
17	Veionella 	Veionella	Veionella

Data Validation	Visualisasi	Aktual	Prediksi
18	Veionella 	Veionella	Veionella
19	Veionella 	Veionella	Veionella
20	Veionella 	Veionella	Veionella

Data Validation	Visualisasi	Aktual	Prediksi
21	Veionella 	Veionella	Veionella

Perbandingan performa arsitektur *inceptionv3* dengan tujuh arsitektur lain yaitu *googlenet*, *vgg16*, *vgg19*, *resnet101*, *inceptionresnetv2*, *squeezenet* dan *xception* dapat dilihat pada Tabel 4.8.

Tabel 4.8 Perbandingan Akurasi Validasi Inceptionv3 dengan Tujuh Arsitektur Lain

No	Arsitektur	Akurasi
1	Inceptionv3	96.97%
2	Googlenet	90.91%
3	Vgg16	42.42%
4	Vgg19	72.73%
5	Resnet101	93.94%
6	Inceptionresnetv2	66.67%
7	Squeezenet	18.18%
8	xception	75.76%

BAB V

PENUTUP

5.1. Kesimpulan

Dari hasil analisis percobaan - percobaan yang telah dilakukan dalam pengujian sistem untuk klasifikasi bakteri dengan *deep learning*, dapat disimpulkan bahwa dari keempat percobaan yang telah dilakukan, sistem klasifikasi bakteri dengan arsitektur *InceptionV3* dengan *k-fold cross validation* dan *augmentation* memiliki rata-rata akurasi tertinggi. Sistem ini mampu mengklasifikasikan lima jenis bakteri yaitu *Acinetobacter baumannii*, *Escherichia coli*, *Neisseria gonorrhoeae*, *Propionibacterium acnes* dan *Veionella*. Hasil percobaan sistem klasifikasi bakteri dengan arsitektur *InceptionV3* dengan *k-fold cross validation* dan *augmentation* menggunakan konfigurasi *training* minibatchsize 5, maxepoch 5, initiallearnrate 0,0001, dan validationfrequency 3 mendapatkan nilai rata-rata akurasi *validation* sebesar 94.43% dengan rata-rata waktu eksekusi selama 44 detik.

5.2. Saran

Sehubungan dengan selesainya penulisan skripsi ini, penulis bermaksud menyampaikan beberapa saran kepada pembaca, yakni:

1. Sistem yang dibuat dapat dikembangkan dengan menambahkan jenis bakteri yang diklasifikasi agar dapat mengklasifikasi lebih banyak jenis bakteri.
2. Sistem yang telah dibuat juga dapat dimodifikasi untuk mengklasifikasi objek lain untuk mengatasi permasalahan lain.

DAFTAR PUSTAKA

Asif, S. *et al.* (2020) ‘Classification of COVID-19 from Chest Radiography Images Using Deep Convolutional Neural Network’, *Journal of Xidian University*, 14(8). doi: 10.37896/jxu14.8/061.

Beale, M. H., Hagan, M. T. and Demuth, H. B. (2020) ‘Deep Learning Toolbox™ User ’ s Guide How to Contact MathWorks’.

Britannica, T. E. of E. (2018) *Bacterial disease*, *Encyclopædia Britannica*. Available at: <https://www.britannica.com/science/bacterial-disease> (Accessed: 9 November 2020).

Choudari, P. (2019) *CNN, Understanding “convolution” operations in*. Available at: <https://medium.com/analytics-vidhya/understanding-convolution-operations-in-cnn-1914045816d4> (Accessed: 12 November 2020).

Gu, S., Pednekar, M. and Slater, R. (2019) ‘Improve Image Classification Using Data Augmentation and Neural Networks’, *SMU Data Science Review*, 2(2), pp. 1–43. Available at: <https://scholar.smu.edu/datasciencereviewhttp://digitalrepository.smu.edu.Availableat:https://scholar.smu.edu/datasciencereview/vol2/iss2/1>.

Khan, S. *et al.* (2018) ‘A Guide to Convolutional Neural Networks for Computer Vision’, *Synthesis Lectures on Computer Vision*, 8(1), pp. 1–207. doi: 10.2200/s00822ed1v01y201712cov015.

Khandelwal, R. (2018) *K fold and other cross-validation techniques*. Available at: <https://medium.com/datadriveninvestor/k-fold-and-other-cross-validation-techniques-6c03a2563f1e> (Accessed: 14 November 2020).

Kurama, V. (2020) *A Review of Popular Deep Learning Architectures: ResNet, InceptionV3, and SqueezeNet*. Available at: <https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/> (Accessed: 13 November 2020).

Malik, F. (2019) *Neural Networks Bias And Weights*. Available at: <https://medium.com/fintechexplained/neural-networks-bias-and-weights-10b53e6285da> (Accessed: 19 January 2021).

Mohamed, B. A. and Afify, H. M. (2019) ‘Automated classification of Bacterial Images extracted from Digital Microscope via Bag of Words Model’, *2018 9th Cairo International Biomedical Engineering Conference, CIBEC 2018 - Proceedings*. IEEE, pp. 86–89. doi: 10.1109/CIBEC.2018.8641799.

Putra, J. W. G. (2019) ‘Pengenalan Konsep Pembelajaran Mesin dan Deep Learning’, 4(August), pp. 1–235.

Putri, M. H. (2017) *Mikrobiologi*.

Rayed Bin Wahed, M. (2016) ‘Comparative Analysis between Inception-V3 and Other Learning Systems using Facial Expressions Detection’, pp. 1–48.

Schleifer, K. H. (2009) ‘Classification of Bacteria and Archaea: Past, present and future’, *Systematic and Applied Microbiology*. Elsevier, 32(8), pp. 533–542. doi: 10.1016/j.syapm.2009.09.002.

Selvaraju, R. R. *et al.* (2019) ‘Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization’, *International Journal of Computer Vision*, 128(2), pp. 336–359. doi: 10.1007/s11263-019-01228-7.

Shaikh, R. (2018) *Cross Validation Explained: Evaluating estimator performance*. Available at: <https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85> (Accessed: 14 November 2020).

Sujaya, N. (2016) *Penuntun Praktikum Mikrobiologi*.

Szegedy, C. *et al.* (2016) ‘Rethinking the Inception Architecture for Computer Vision’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

Talo, M. (2019) ‘An Automated Deep Learning Approach for Bacterial Image Classification’, pp. 1–5. Available at: <http://arxiv.org/abs/1912.08765>.

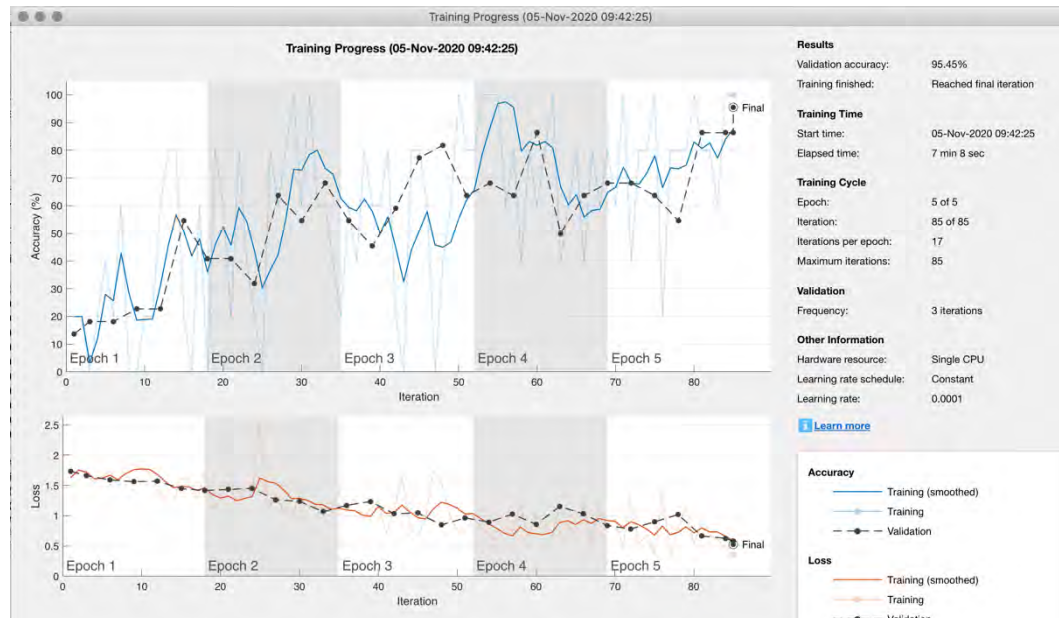
Tsang, S.-H. (2018) *Review: Inception-v3 — 1st Runner Up (Image Classification) in ILSVRC 2015*. Available at: <https://sh-tsang.medium.com/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c> (Accessed: 13 November 2020).

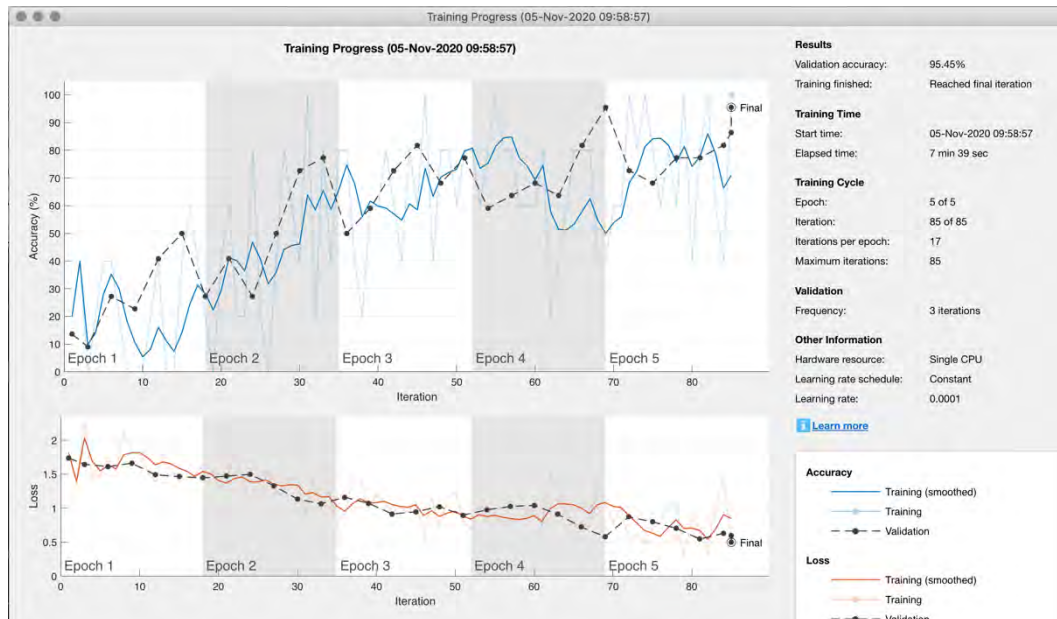
Wang, C. *et al.* (2019) ‘Pulmonary image classification based on inception-v3 transfer learning model’, *IEEE Access*. IEEE, 7, pp. 146533–146541. doi: 10.1109/ACCESS.2019.2946000.

Zieliński, B. *et al.* (2017) ‘Deep learning approach to bacterial colony classification’, *PLoS ONE*, 12(9). doi: 10.1371/journal.pone.0184554.

LAMPIRAN

Lampiran 1 Graph Pelatihan Arsitektur *InceptionV3* dengan *K-fold cross validation*

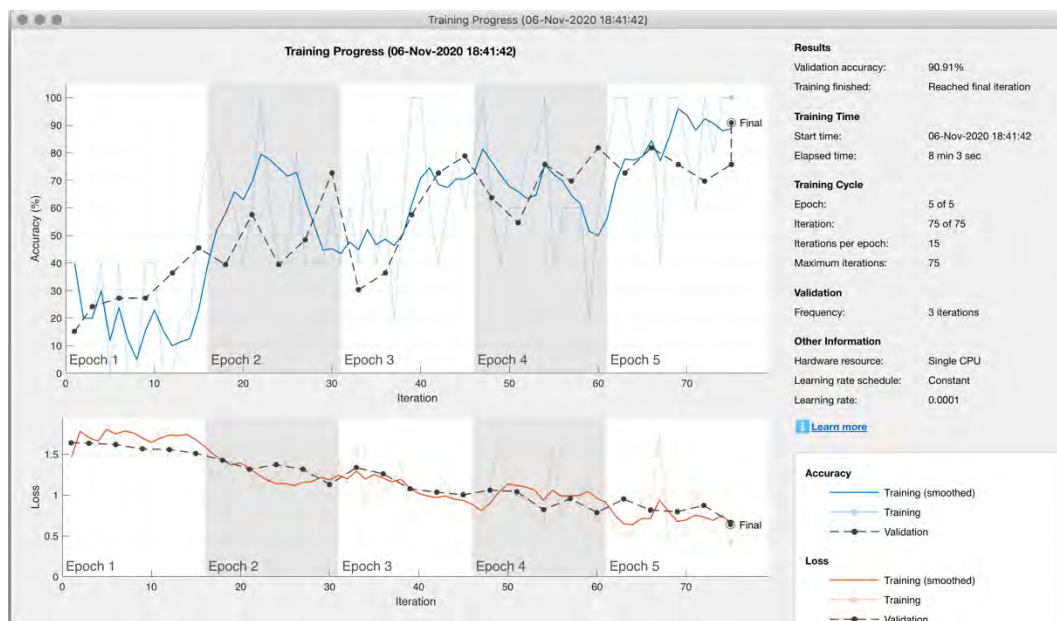
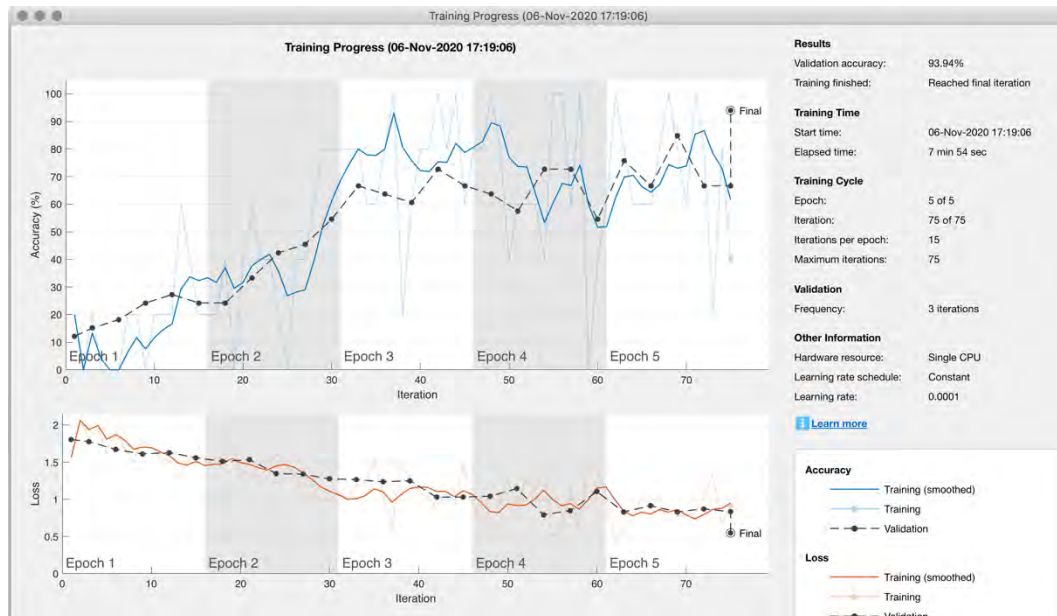


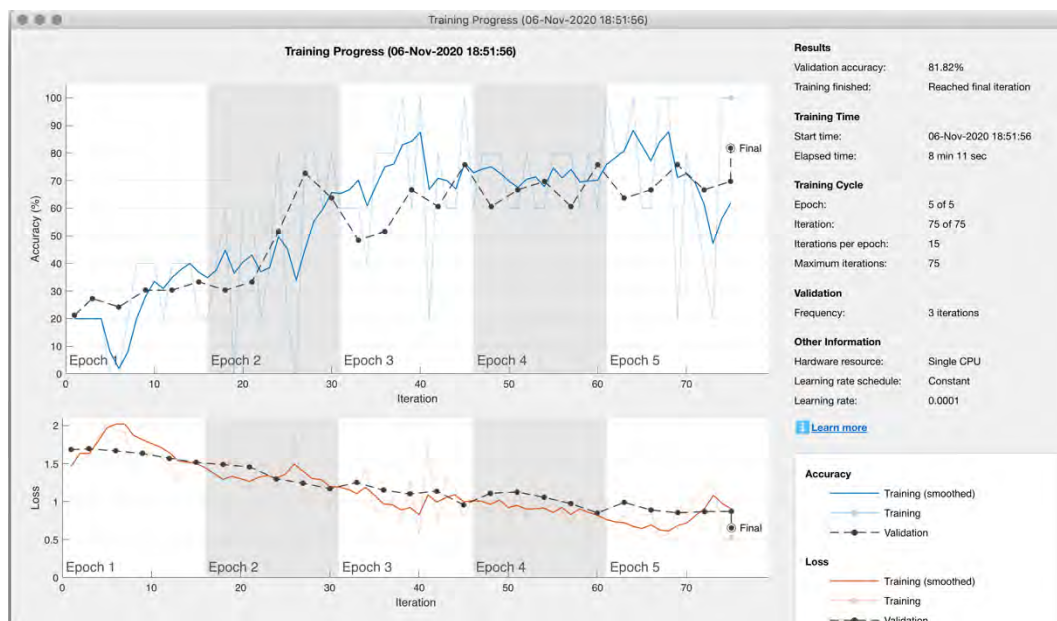
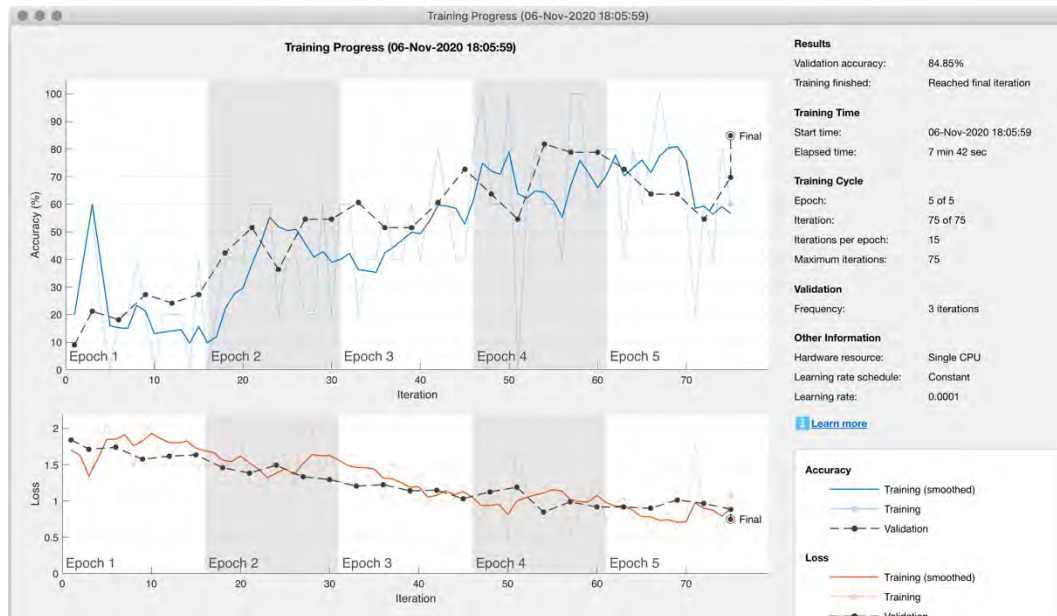




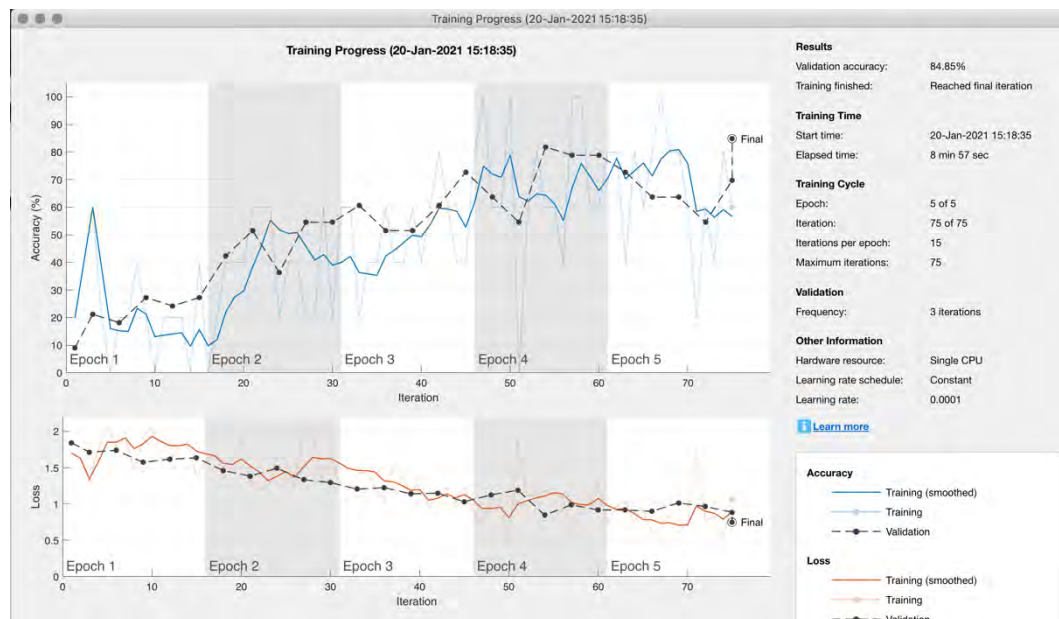
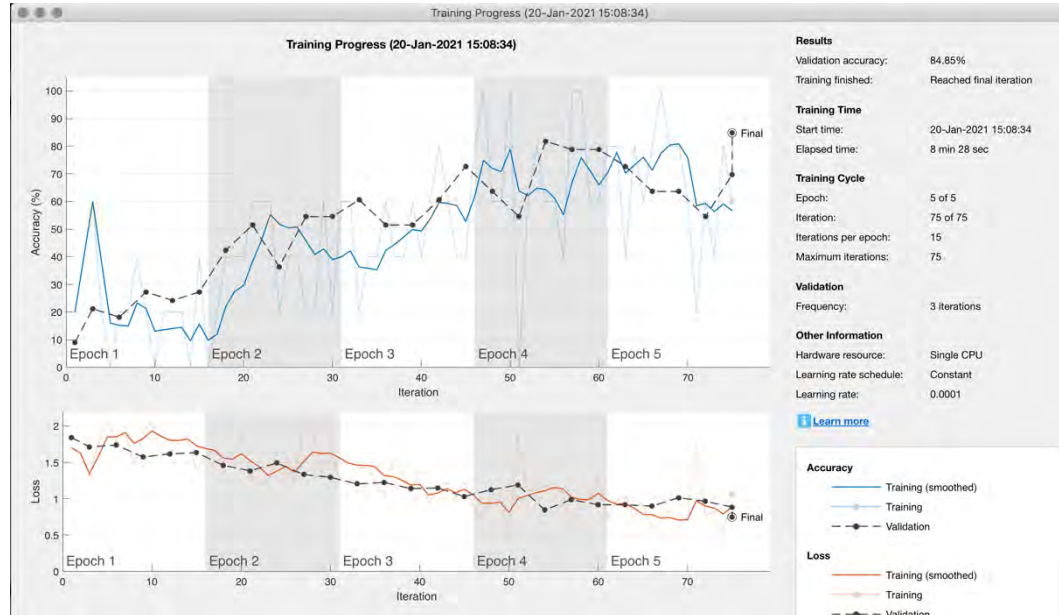
Lampiran 2 *Graph Pelatihan Arsitektur InceptionV3 tanpa K-fold cross validation*

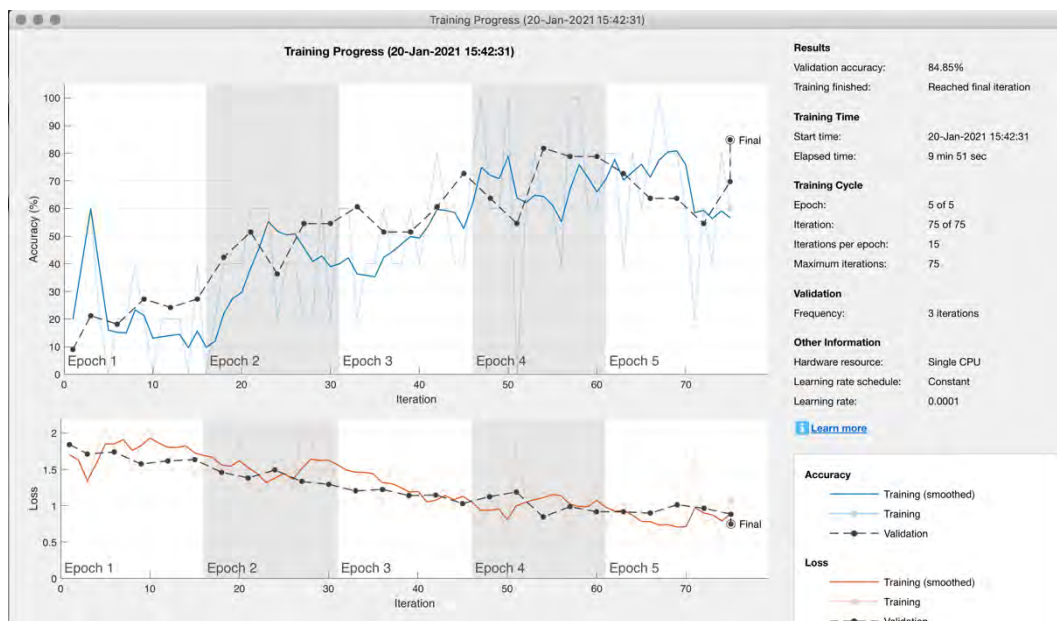
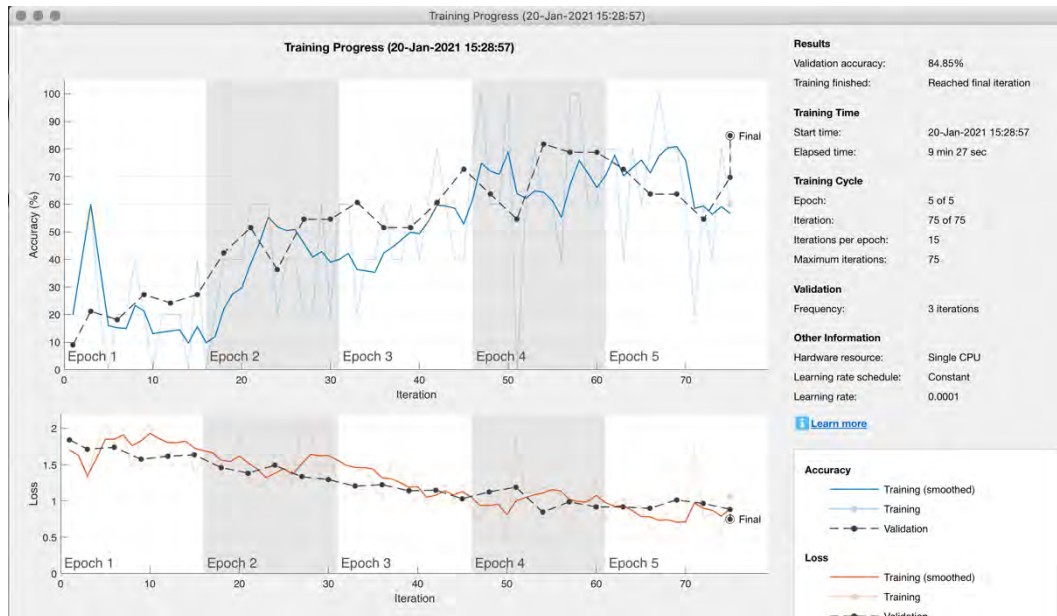


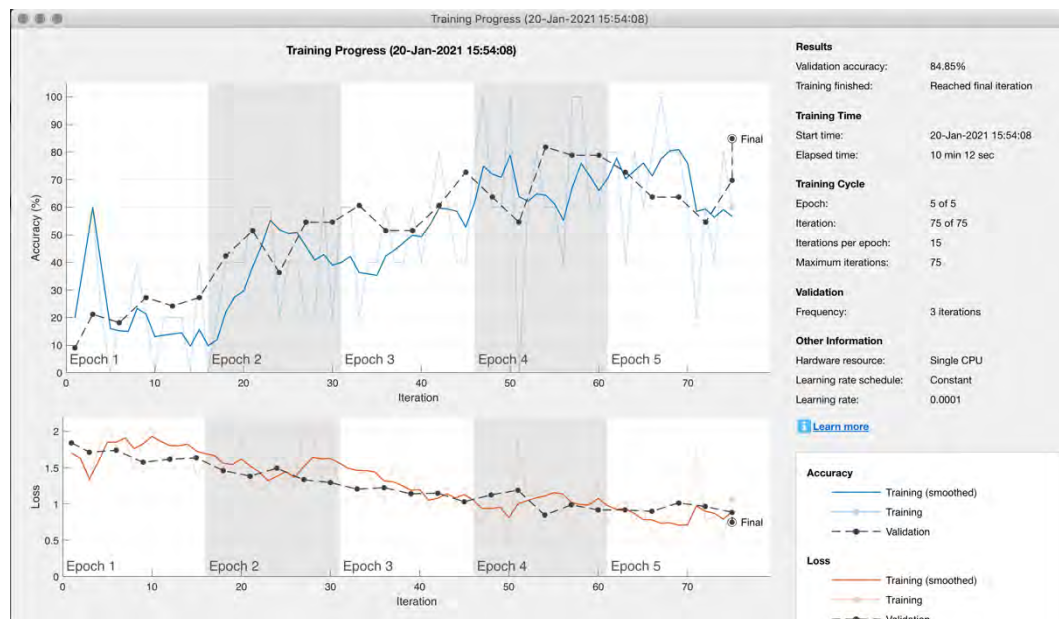




Lampiran 3 *Graph* Pelatihan Arsitektur *InceptionV3* dengan konfigurasi *RNG* default

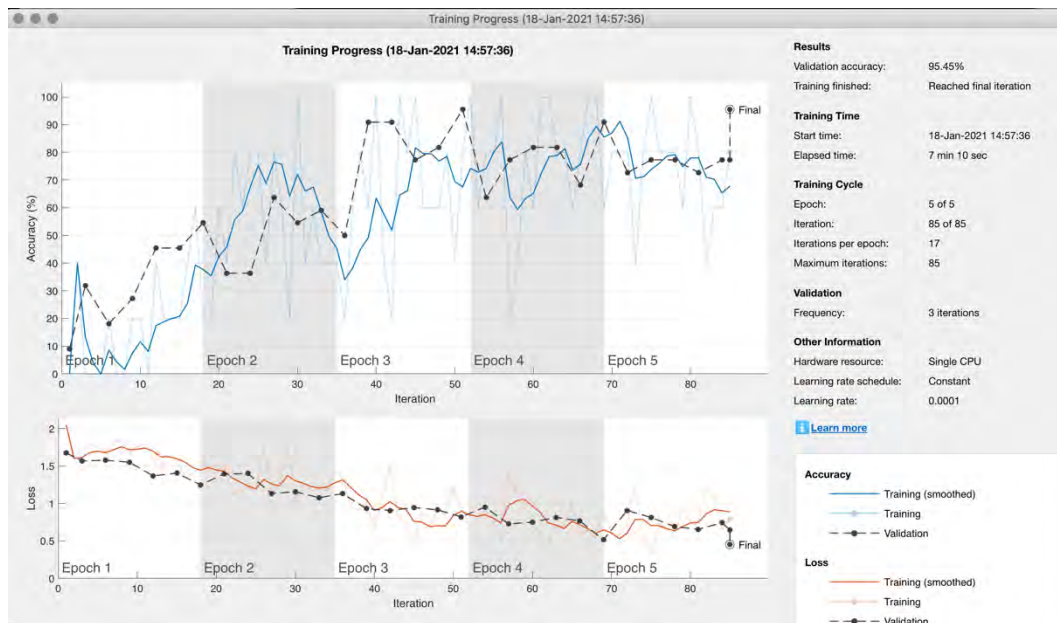
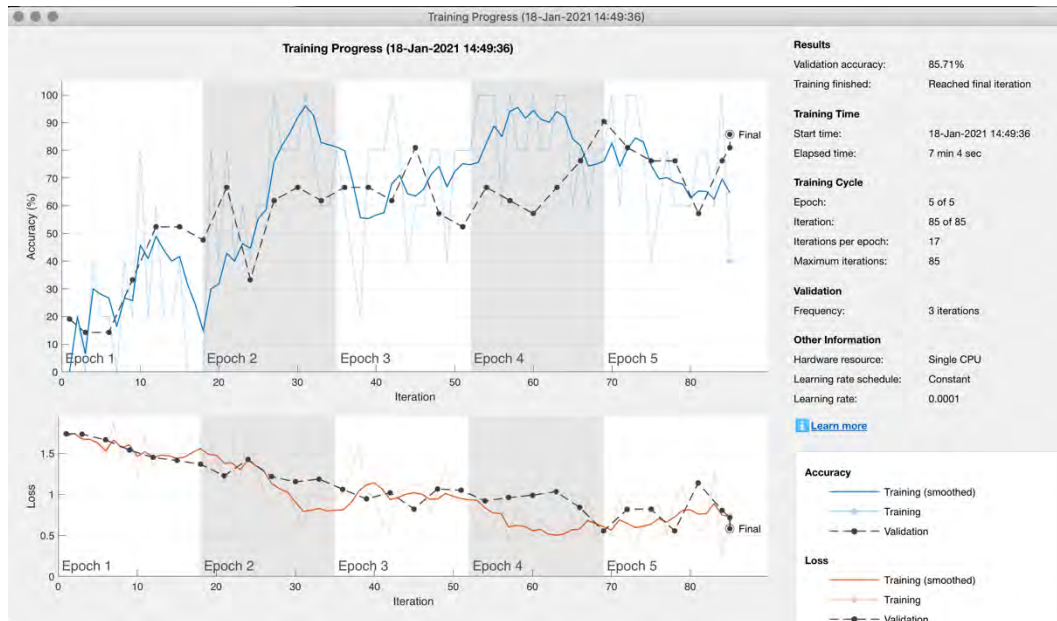


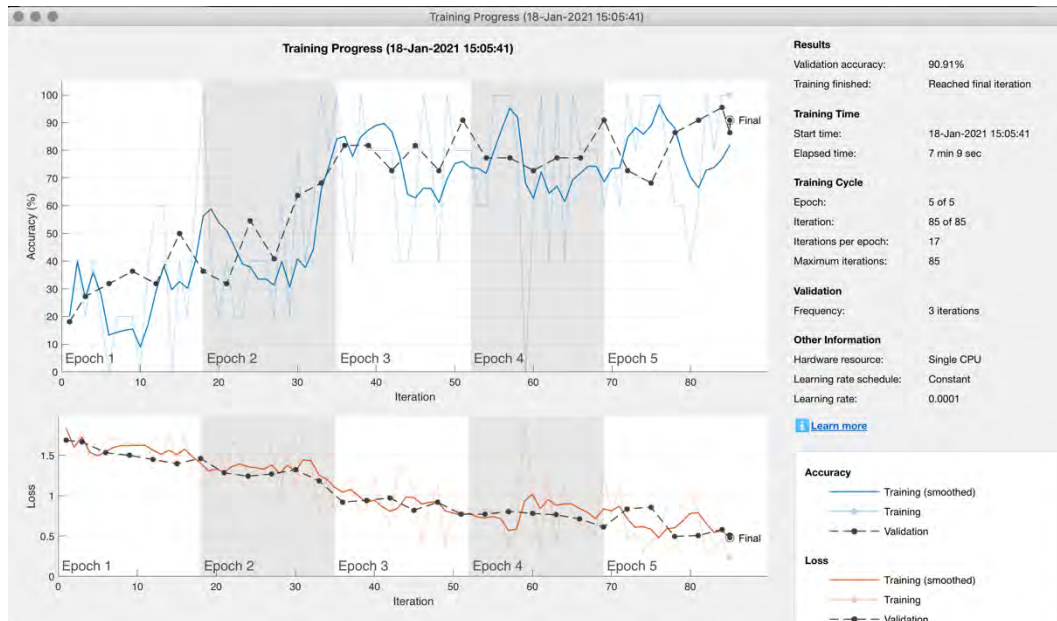




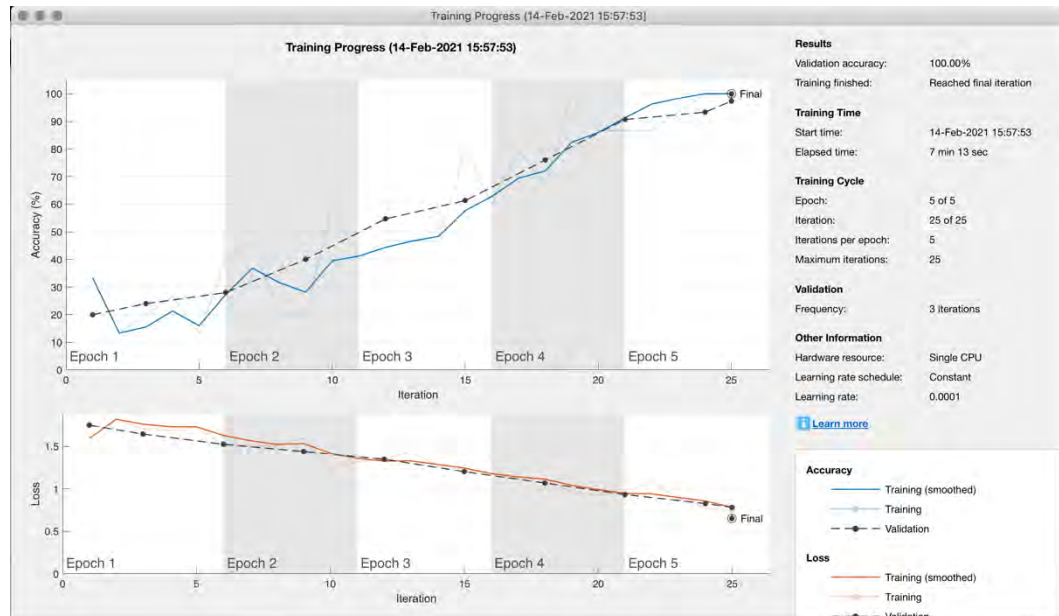
Lampiran 4 *Graph* Pelatihan Arsitektur *InceptionV3* dengan *k-fold cross validation* tanpa augmentation.





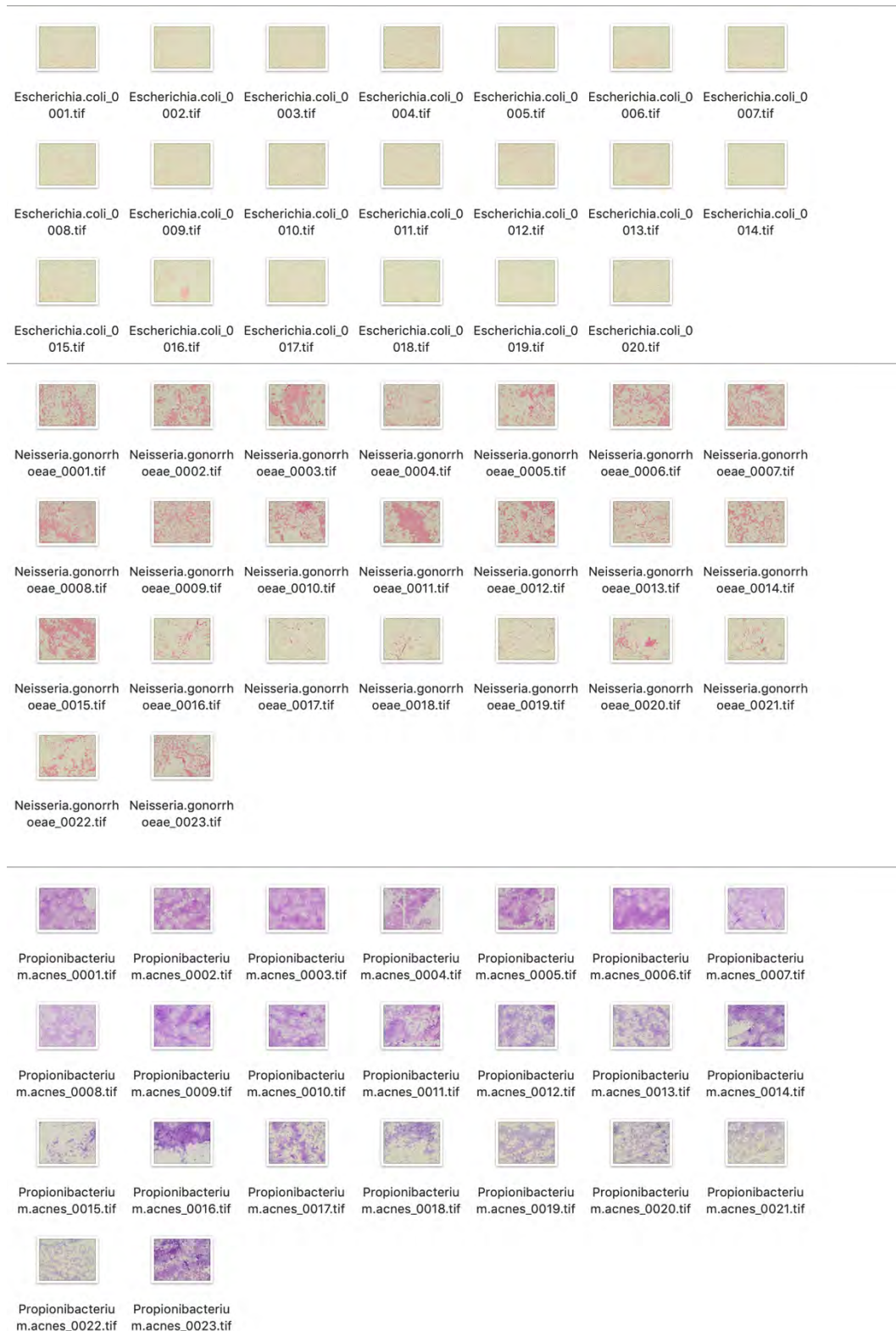


Lampiran 5 *Graph* Pelatihan Percobaan Menggunakan Data *Training* sebagai data validasi.



Lampiran 6 Data Citra *Training* dan *Validation*







LEMBAR PERBAIKAN SKRIPSI

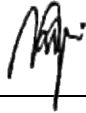
“KLASIFIKASI BAKTERI DENGAN METODE DEEP LEARNING”

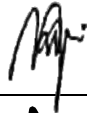
OLEH:

ALIF TRI HANDOYO
D42116504

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 1 Februari 2021.
Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Dr. Ir. Ingrid Nurtanio, M.T.	
Sekretaris	Anugrayani Bustamin, S.T., M.T.	
Anggota	Dr.Eng. Intan Sari Areni, S.T., M.T.	
	Elly Warni, S.T, M.T	

Pembimbing	Nama	Tanda Tangan
I	Dr. Ir. Ingrid Nurtanio, M.T.	
II	Anugrayani Bustamin, S.T., M.T.	



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN
No. 1468/UN4.7.1/TD.06/2020

Dari : Dekan Fakultas Teknik Universitas Hasanuddin

Kepada : 1. Dr. Ir. Ingrid Nurtanio, M.T Pemb. I
2. Anugrayani Bustamin, ST., M.T Pemb. II

Isi : 1. Berdasarkan Surat Ketua Departemen Teknik Informatika Fakultas Teknik Nomor. 078/UN4.7.7./TD.06/2020 tanggal 27 Januari 2020 tentang usul DOSEN PEMBIMBING MAHASISWA, maka dengan ini kami menugaskan Saudara untuk membimbing penulisan Skripsi/Tugas Akhir mahasiswa Teknik Informatika Fakultas Teknik Universitas Hasanuddin di bawah ini :

N a m a : No. Stambuk :
Alif Tri Handoyo D421 16 504

Judul Skripsi/Tugas Akhir :

“ Klasifikasi dan Perhitungan Jumlah Bakteri dengan Metode Deep Learning “

2. Surat penugasan pembimbing ini mulai berlaku sejak tanggal ditetapkannya dan berakhir sampai selesainya penulisan Skripsi/Tugas Akhir mahasiswa tersebut.
3. Agar penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.

Ditetapkan di Gowa
Pada tanggal 27 Januari 2020

 a.n. Dekan,
Wakil Dekan Bidang Akademik


Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
NIP. 196903081995121001

Tembusan :

1. Dekan FT-UH,
2. Ketua Departemen Teknik Informatika FT-UH,
3. Mahasiswa yang bersangkutan





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

SURAT KETERANGAN NILAI UJIAN SEMINAR PROPOSAL

Nomor : 121.../UN4.7.7.TI/DA.04.09/2020

Pada hari ini **Rabu**, tanggal 5 **Februari** Pukul **09.00 WITA-Selesai** bertempat di **Lab. Ubicon Departemen Teknik Informatika**, telah dilaksanakan Ujian Seminar Proposal bagi Saudara:

Nama : Alif Tri Handoyo

No. Stambuk : D42116504

Fakultas/Departemen : Teknik/Teknik Informatika

Judul Skripsi : **“Klasifikasi dan Perhitungan Jumlah Bakteri dengan Metode Deep Learning “**

Setelah pembawa ujian seminar proposal menguraikan tugas akhirnya dan menjawab pertanyaan dengan dinyatakan lulus/~~tidak lulus~~, Baik/~~Cukup~~/~~Sedang~~.

Maka berdasarkan hasil penilaian dinyatakan lulus / ~~tidak lulus~~

Dengan nilai :

A — B+ — B — B — C+ — C — D — E

Mengetahui:

Ketua Departemen
Teknik Informatika FT-UH,

Dr. Amil Ahmad Ilham, ST., M.IT
Nip. 19731010 199802 1 001

Dosen Penguji,

Dr. Ir. Ingrid Nurtanio, M.T.
Nip. 19610813 198811 2 001

Diketahui oleh,
Dekan,

Wakil Dekan Bidang Akademik, Riset dan Inovasi

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
NIP. 19690308 199512 1 001



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA UJIAN SEMINAR PROPOSAL

Pada hari ini **Rabu**, tanggal **5 Februari** Pukul **09.00 WITA-Selesai** bertempat di **Lab. Ubicon Departemen Teknik Informatika**, telah dilaksanakan Ujian Seminar Proposal bagi Saudara:

Nama : Alif Tri Handoyo
No. Stambuk : D42116504
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : **“Klasifikasi dan Perhitungan Jumlah Bakteri dengan Metode Deep Learning”**

Yang dihadiri oleh panitia Ujian Seminar Proposal sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr. Ir. Ingrid Nurtanio, M.T.	Pemb I/Ketua	1...
2.	Anugrayani Bustami, S.T., M.T	Pemb II/Sekretaris	2...
3.	Dr.Eng. Intan Sari Areni, S.T., M.T.	Anggota	3...
4.	Elly Warni, S.T., M.T.	Anggota	4...

Hasil keputusan panitia penilai Ujian Seminar Proposal Tugas Akhir : **Lulus** / ~~Tidak lulus~~ dengan nilai angka*86,75*..... dan huruf*A*.....

Gowa, 5 Februari 2020

Ketua/Sekretaris Panitia Ujian,

Dr. Ir. Ingrid Nurtanio, M.T.



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN
No. 18723/UN4.7.1/TD.06/2020

Dari : Dekan Fakultas Teknik Universitas Hasanuddin
Kepada : Mereka yang tercantum namanya dibawah ini
Isi : 1. Bahwa berdasarkan peraturan Akademik Universitas Hasanuddin Tahun 2003 Pasal 36 butir 3 point a, b (SK. Rektor Unhas Nomor : 1067/J04/PP.08/2008), dengan ini menugaskan Saudara sebagai PANITIA SEMINAR HASIL Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I/ Ketua : 1. Dr. Ir. Ingrid Nurtanio, M.T
Pembimbing II/ Sekretaris : 2. Anugrayani Bustamin, ST., M.T
Anggota : 3. Dr.Eng. Intan Sari Areni, ST., M.T
4. Elly Warni, ST., M.T

Untuk menguji bagi mahasiswa tersebut dibawah ini :

Nama/NIM : Alif Tri Handoyo D421 16 504
Program Studi : Teknik Informatika
Judul thesis/Skripsi : KLASIFIKASI BAKTERI DENGAN METODE
DEEP LEARNING

2. Waktu seminar ditetapkan oleh Panitia Seminar Hasil Program Strata Satu (S1)
3. Agar Surat Penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
4. Surat penugasa ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya seminar tersebut dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudia hari terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa
Pada tanggal 8 Desember 2020
a.n. Dekan,
Wakil Dekan Bidang Akademik

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
NIP. 196903081995121001

Tembusan :

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Mahasiswa yang bersangkutan





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://cs.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

SURAT KETERANGAN NILAI SEMINAR HASIL

Nomor : / UN4.7.7.TI/PK.03.06/2020

Pada hari ini **Kamis**, tanggal **10 Desember 2020** Pukul **10.00 WITA** - **Selesai** bertempat di **Ruang Lab. AIMP Departemen Teknik Informatika**, telah dilaksanakan Seminar Hasil bagi Saudara :

Nama : Alif Tri Handoyo
No. Stambuk : D421 16 504
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : "Klasifikasi Bakteri dengan Metode Deep Learning"

Setelah pembawa seminar hasil menguraikan tugas akhirnya dan menjawab pertanyaan dari Tim Penguji dinyatakan Lulus / Tidak Lulus dengan nilai :

A 06 A B+ B B C+ C D E

Megetahui.

A.n Ketua Departemen Tek.Informatika,
Sekretaris Departemen

Dr. Amil Ahmad Ilham, ST M.I.T
Nip. 19731010 199802 1 001

Dosen Penguji,

Dr. Ir. Ingrid Nurtanio, M.T
Nip. 19610813 198811 2 001

Diketahui oleh,
a.n Dekan,
Wakil Dekan Bidang Akademik, Riset dan Inovasi

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
Nip. 19690308 199512 1 001



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA SEMINAR HASIL

Pada hari ini **Kamis**, tanggal **10 Desember 2020** Pukul **10.00 WITA** - **Selesai** bertempat di **Ruang Lab. AIMP Departemen Teknik Informatika Gowa**, telah dilaksanakan Seminar Hasil bagi Saudara :

Nama : Alif Tri Handoyo
No. Stambuk : D421 16 504
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : "Klasifikasi Bakteri dengan Metode Deep Learning"

Yang dihadiri oleh Tim Penguji Seminar Hasil sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr. Ir. Ingrid Nurtanio,M.T	Pemb I/Ketua	1.
2.	Anugrayani Bustamin,ST.,M.T	Pemb II/Sekretaris	2.
3.	Dr. Eng.Intan Sari Areni,ST.,M.T	Anggota	3.
4.	Elly Warni,ST.,M.T	Anggota	4.

Hasil keputusan Tim Penguji Seminar Hasil : **Lulus** / **Tidak lulus** dengan nilai angka **86**
dan huruf **A**

Makassar, 10 Desember 2020
Ketua/Sekretaris Panitia Ujian,

Dr. Ir. Ingrid Nurtanio,M.T



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

DAFTAR HADIR SEMINAR HASIL

Nama/Stambuk : I. Alif Tri Handoyo D421 16 504

Judul Skripsi/T.A : "Klasifikasi Bakteri dengan Metode Deep Learning"

Hari/Tanggal : Kamis, 10 Desember 2020

Jam : 10- 00 Wita – Selesai

Tempat : Ruang Lab.AIMP Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
I.	Pembimbing I	1. Dr. Ir. Ingrid Nurtanio, M.T	1.
	Pembimbing II	2. Anugrayani Bustamin, ST., M.T	2.
II.	Anggota Penguji	3. Dr. Eng. Intan Sari Areni, ST., M.T	3.
		4. Elly Warni, St., M.T	4.

PANITIA UJIAN

Ketua,

Dr. Ir. Ingrid Nurtanio, M.T

Sekretaris,

Anugrayani Bustamin, ST., M.T



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN
No. 975/UN4.7.1/TD.06/2021

Dari : Dekan Fakultas Teknik Universitas Hasanuddin.

Kepada : Mereka yang tercantum namanya di bawah ini.

I s i : 1. Bahwa berdasarkan peraturan Akademik Universitas Hasanuddin Tahun 2018 pasal 18 butir point a, b (SK.Rektor Unhas nomor : 2781/UN4.1/KEP/2018), dengan ini menugaskan Saudara sebagai PANITIA UJIAN SARJANA Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I / Ketua : 1. Dr. Ir. Ingrid Nurtanio, M.T
Pembimbing II/ Sekretaris : 2. Anugrayani Bustamin, ST., M.T
Anggota : 3. Dr.Eng. Intan Sari Areni, ST., M.T
4. Elly Warni, ST., M.T

untuk menguji bagi mahasiswa tersebut di bawah ini :

Nama/NIM : Alif Tri Handoyo D421 16 504
Program Studi : Teknik Informatika
Judul Thesis/Skripsi : **Klasifikasi Bakteri dengan Metode Deep Learning**

2. Waktu Ujian ditetapkan oleh Panitia Ujian Sarjana Program Strata Satu (S1).
3. Agar Surat penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
4. Surat penugasan ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya Ujian Sarjana tersebut, dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudian hari ternyata terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa,
Pada tanggal 26 Januari 2021
a.n. Dekan

Wakil Dekan Bidang Akademik

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
NIP.1969030819951210011

Tembusan :

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Kasubag. Umum dan Perlengkapan FT-UH





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
Email : informatika@unhas.ac.id

SURAT KETERANGAN NILAI UJIAN SKRIPSI

Nomor : / UN4.7.7.TI/PK.03.06/2021

Pada hari ini **Senin**, tanggal **01 Februari 2021** Pukul **10.00 WITA** - **Selesai** bertempat di **Ruang Lab. AIMP Departemen Teknik Informatika Gowa**, telah dilaksanakan Ujian Skripsi bagi Saudara :

Nama : Alif Tri Handoyo
No. Stambuk : D421 16 504
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : **"Klasifikasi Bakteri dengan Metode Deep Learning"**

Setelah pembawa ujian Skripsi menguraikan tugas akhirnya dan menjawab pertanyaan dari Tim Penguji dinyatakan Lulus / Tidak Lulus dengan nilai :

A **87** A- B+ B B- C+ C D E

Mengetahui:

Ketua Departemen Tek.Informatika,

Dosen Penguji,

Dr. Amil Ahmad Ilham, ST M.I.T
Nip. 19731010 199802 1 001

Dr. Ir.Ingrid Nurtanio, M.T
Nip. 19610813 198811 2 001

Diketahui oleh,
a.n. Dekan,
Wakil Dekan Bidang Akademik, Riset dan Inovasi

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
Nip. 19690308 199512 1 001



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
Email : informatika@unhas.ac.id

BERITA ACARA UJIAN SKRIPSI

Pada hari ini **Senin** tanggal **01 Februari, 2021** Pukul **10.00 WITA** - **Selesai** bertempat di **Ruang Lab. AIMP Departemen Teknik Informatika Gowa**, telah dilaksanakan Ujian Skripsi bagi Saudara :

Nama : Alif Tri Handoyo
No. Stambuk : D421 16 504
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : **“Klasifikasi Bakteri dengan Metode Deep Learning”**

Yang dihadiri oleh Tim Penguji Ujian Skripsi sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr. Ir.Ingrid Nurtanio, M.T	Pemb I/Ketua	1.....
2.	Anugrayani Bustamin,ST.,M.T	Pemb II/Sekretaris	2.....
3.	Dr.Eng.Intan Sari Areni,ST.,M.T	Anggota	3.....
4.	Elly Warni,ST.,M.T	Anggota	4.....

Hasil keputusan Tim Penguji Ujian Skripsi/Tugas Akhir : **Lulus / Tidak lulus** dengan nilai angka **87** dan huruf **A**

Gowa, 01 Februari 2021

Ketua/Sekretaris Panitia Ujian,

Dr. Ir.Ingrid Nurtanio, M.T



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
Email : informatika@unhas.ac.id

**DAFTAR HADIR UJIAN SKRIPSI MAHASISWA
FAKULTAS TEKNIK UNHAS**

Nama/Stambuk : 1. Alif Tri Handoyo D421 16 504

Judul Skripsi/T.A : "Klasifikasi Bakteri dengan Metode Deep Learning"

Hari/Tanggal : Senin, 01 Februari 2021

Jam : 10.00 Wita – Selesai

Tempat : Ruang Lab. AIMP Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Dr. Ir. Ingrid Nurtanio, M.T	1
	Pembimbing II	2. Anugrayani Bustamin, ST., M.T	2
II.	Anggota Penguji	3. Dr. Eng. Intan Sari Areni, ST., M.T	3
		4. Elly Warni, ST., M.T	4

PANITIA UJIAN

Ketua,

Sekretaris,

Dr. Ir. Ingrid Nurtanio, M.T

Anugrayani Bustamin, ST., M.T